

Kreatives Hardware Hacking



mit der Arduino Plattform



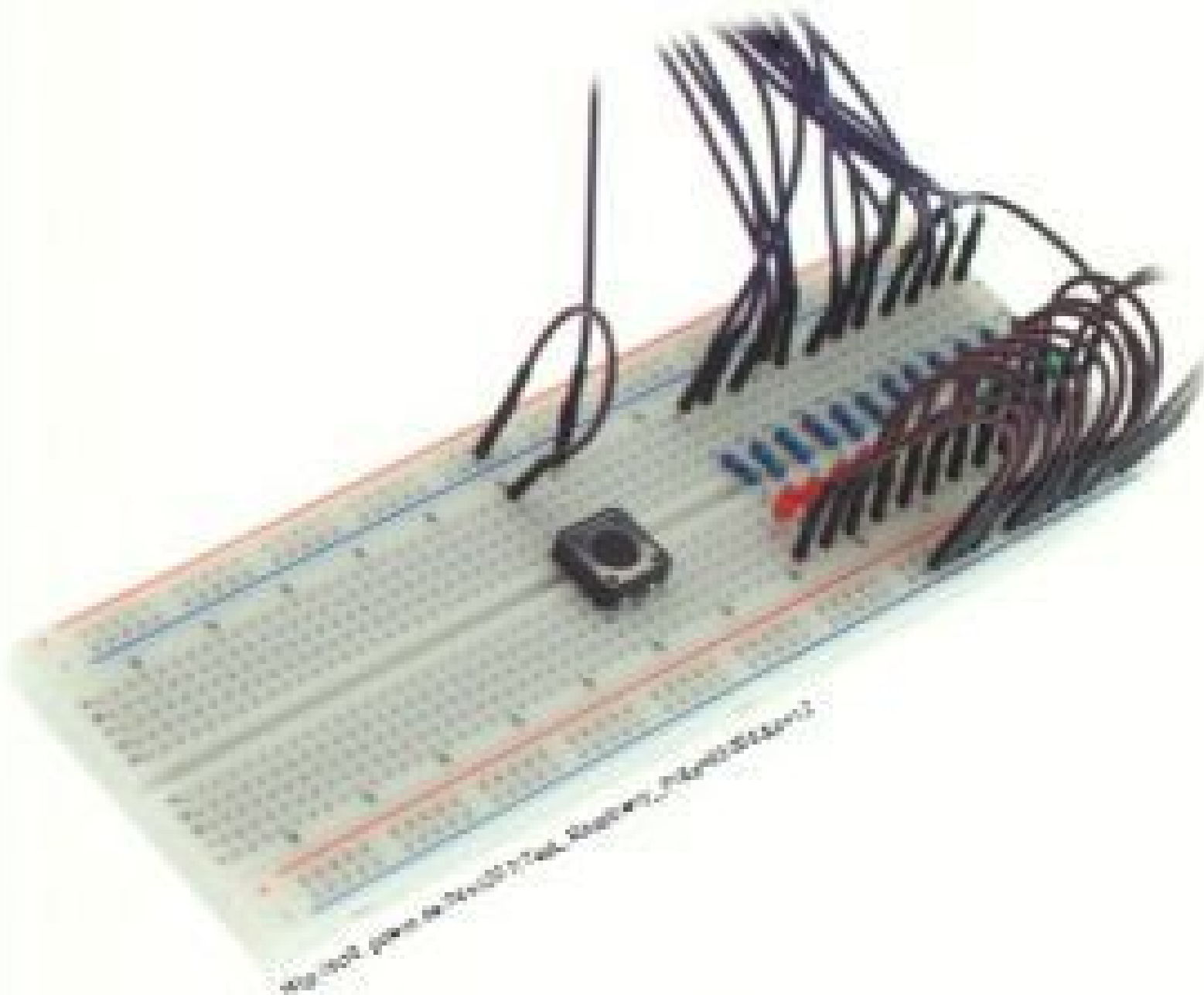
Stefan Wendler und Martin Förtsch

14.06.13, München

Agenda

- **Einführung in die Arduino-Plattform**
- Was ist das – MIDI?
- Beat Sneakers
- Laser Theremin
- Lavalampe
- Der menschliche Körper als Controller LEAP

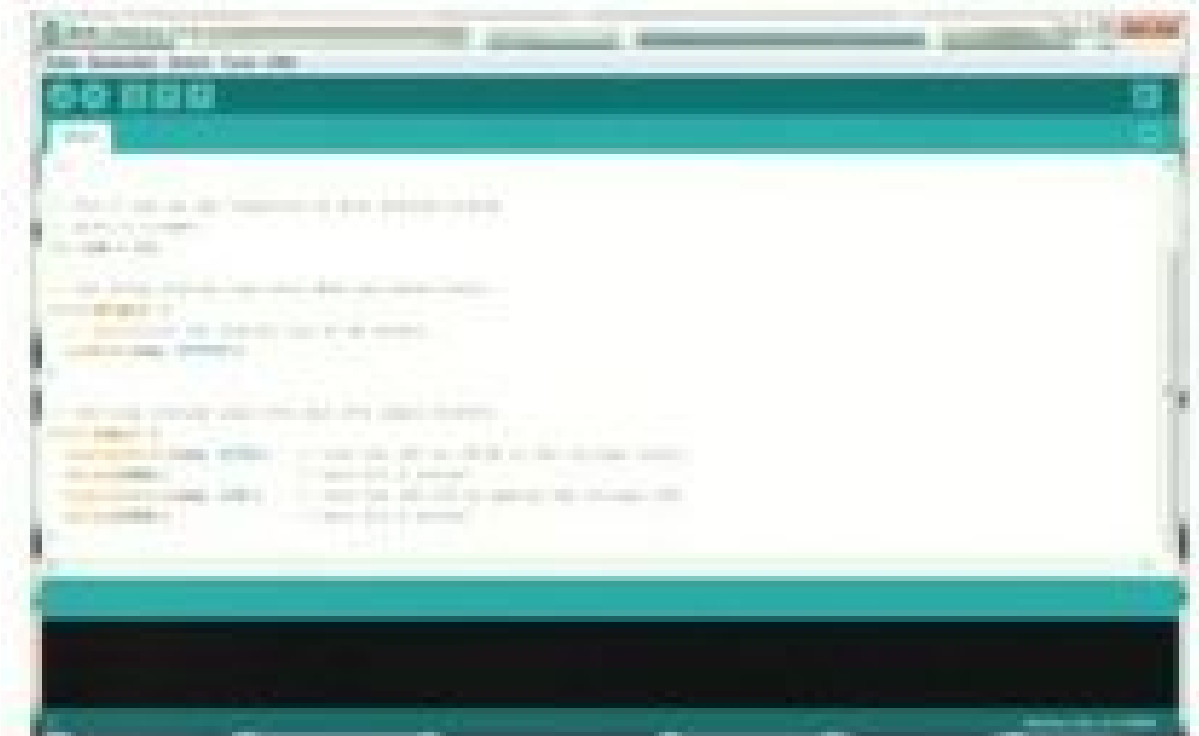
Was ist Arduino?



**Physical computing Platform
für Elektronik-Prototyping**



Open Source Hardware



Open Source Software

Was ist Arduino?

```
void loop() {  
  Serial.println((int) analogRead(PITCHINPUT));  
  delay(LOOPDELAY);  
}
```

Wiring-basierte Programmiersprache
(vergleichbar mit C/C++)



USB-programmierbar



<http://www.arduino.cc>

große Community

Arduino Uno

- AVR ATmega 328 Mikrocontroller
- 16 kByte EEPROM
- 1 kByte RAM
- 16 MHz Clock
- Ein- und Ausgänge
 - 14 digitale Ein- und Ausgänge
 - 6 Analog Eingänge
 - 6 PWM-Ausgänge
 - I²C-Bus, Serial Bus (TX/RX)



Arduino Uno

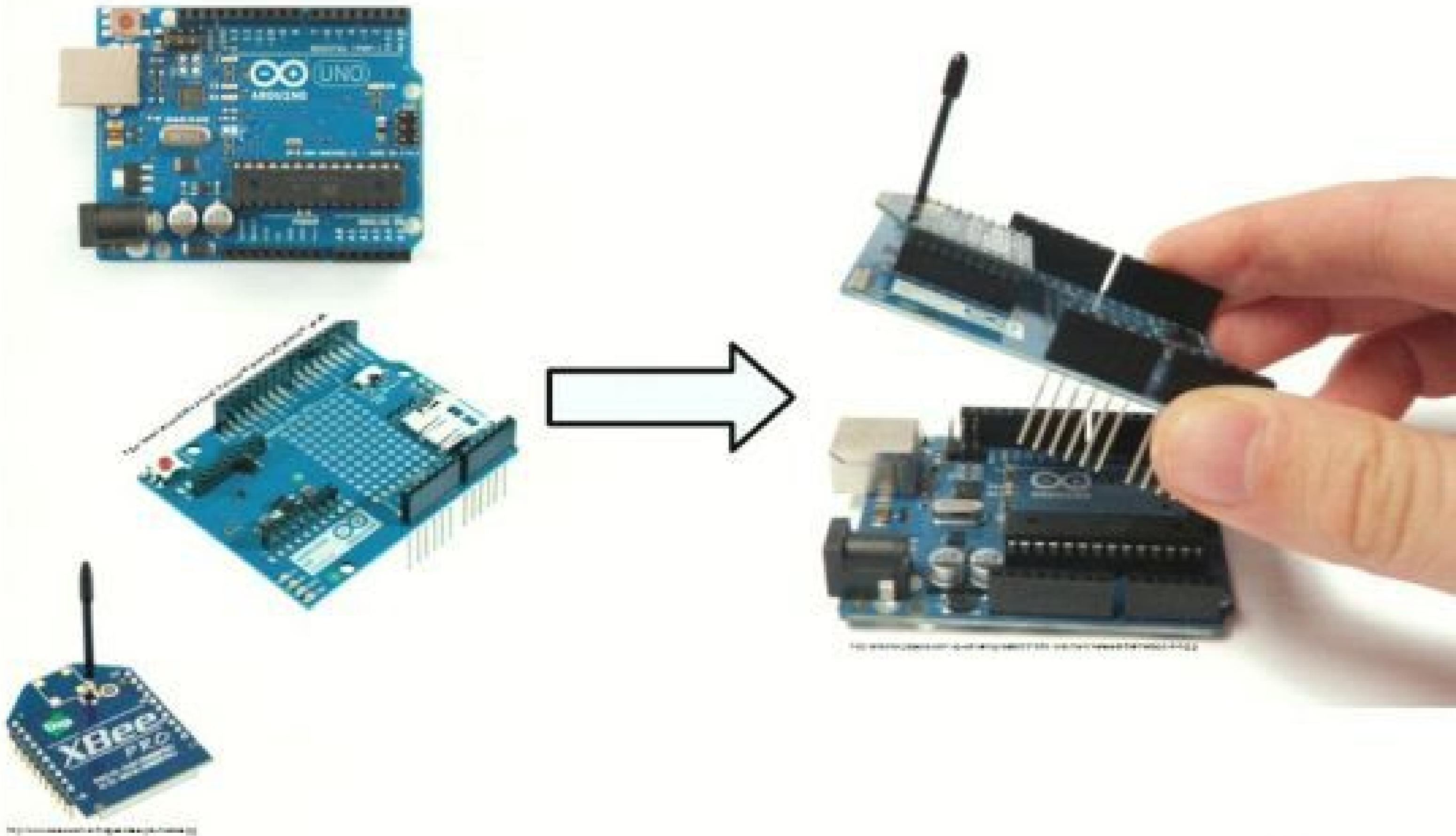


Viele weitere Modelle und Größen



<http://www.arduino.cc/en/uploads/media/arduino-uno-rev31.jpg>

Arduino-Shields



Arduino-Shields



Wireless SD Shield



Ethernet-Shield



GPS-Shield



GSM/GPRS-Shield

Projektideen



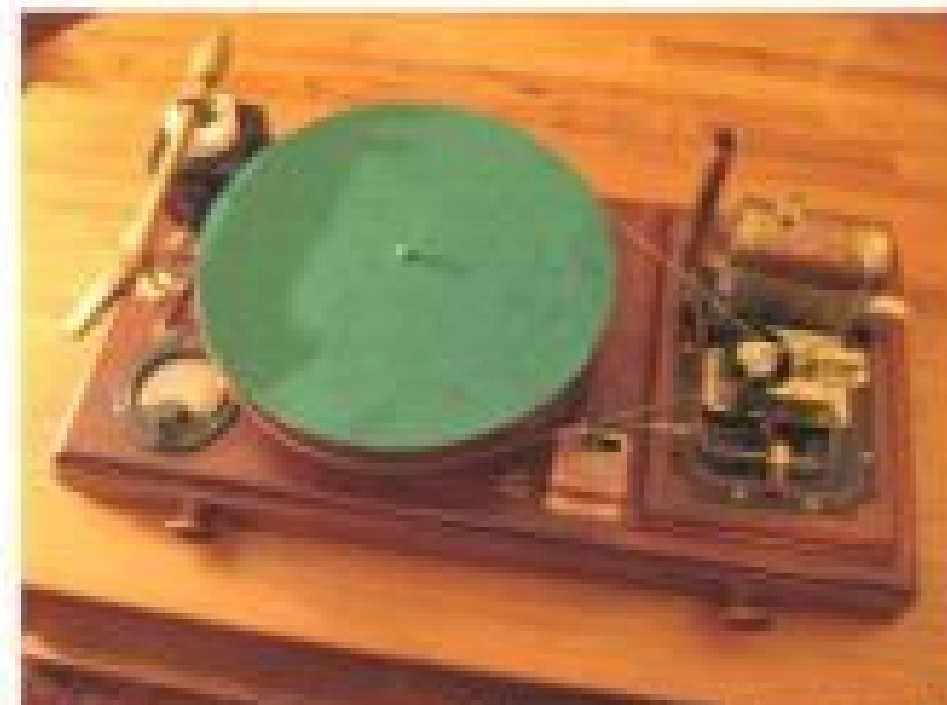
<http://www.instructables.com/blog/water-harp-laser-harp-2015-02-10/1001483/>

Laserharp



<http://www.airpiano.com/airpiano-downloads/2015/11/airpiano1.jpg>

Airpiano



<http://www.hacking.com/content/tmg-01-steampunk-record-player-with-arduino-1.jpg>

Steampunk Plattenspieler

Warum nicht Raspberry Pi?

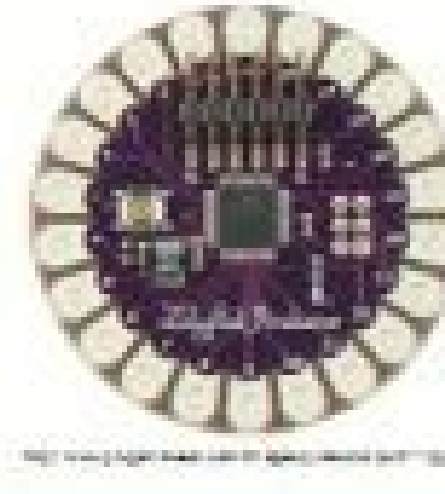
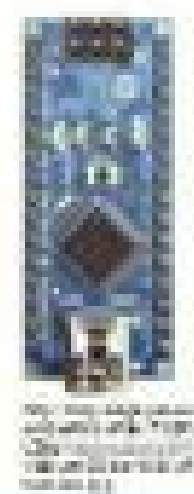


<https://www.raspberrypi.org/documentation/hardware/raspberrypi/raspberrypi1-model-b-plus-100-pin-headers.html>

Anforderung an Musikinstrumente

- Keine bzw. extrem niedrige Latenzen
- Schnelle Einschaltzeiten
- Je nach Bauweise versch. Controller-Größen
- sehr niedriger Energieverbrauch
 - erspart meist zusätzliche Stromkabel

Arduino vs. Raspberry



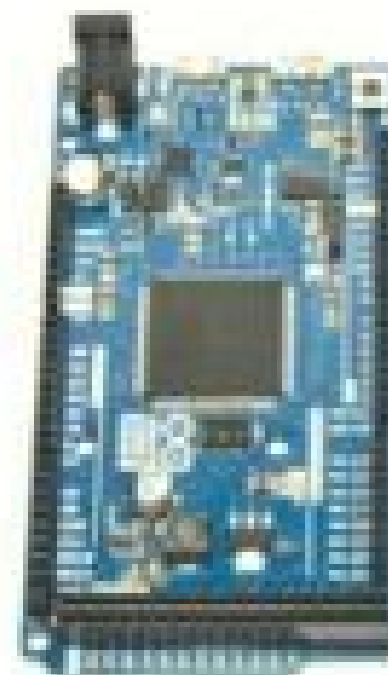
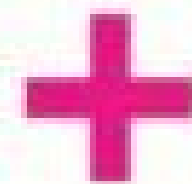
	Arduino	Raspberry Pi
Prozessor	Mikrocontroller	Anwendungsprozessor
Ausführungen	viele versch. Baugrößen	eine Baugröße
PWM-Ausgabeports	Ja (z.B. Uno: 6 Ports)	nur einen
Latenzen	extrem niedrig (<5ms)	hoch (>20ms) (OS-Overhead)
Einsatzgebiet	Arbeiten mit (komplexen) elektronischen Schaltungen	Advanced Software und einfache Hardwaresteuerung
Preis	Uno Rev 3 ca. 20 € Nano Rev 3 ca. 20 €	Model A ca. 25 € Model B ca. 35 €

Weitere Plattformen

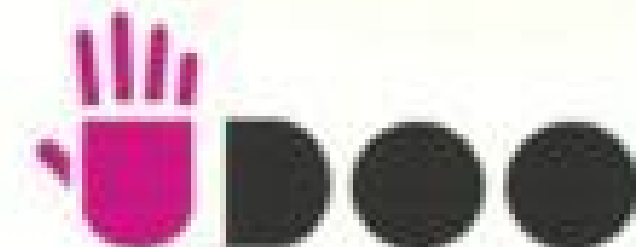


4xRaspberry Pi

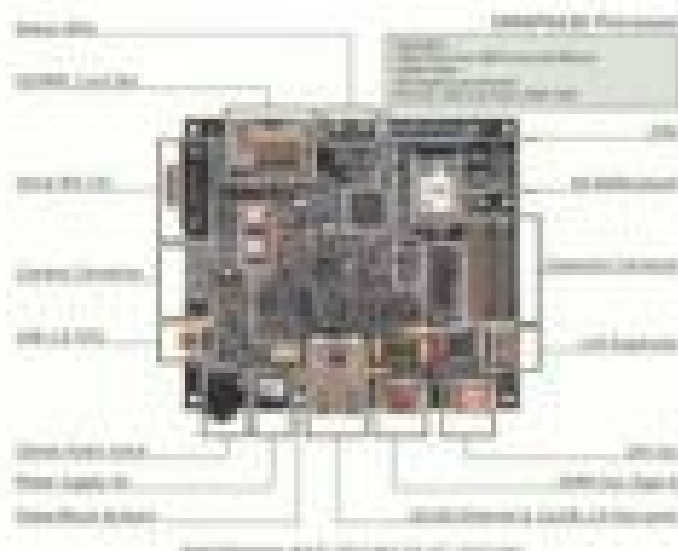
<http://www.raspberrypi.org/documentation/hardware/raspberrypi/3b-plus-board-01.pdf>



Arduino DUE



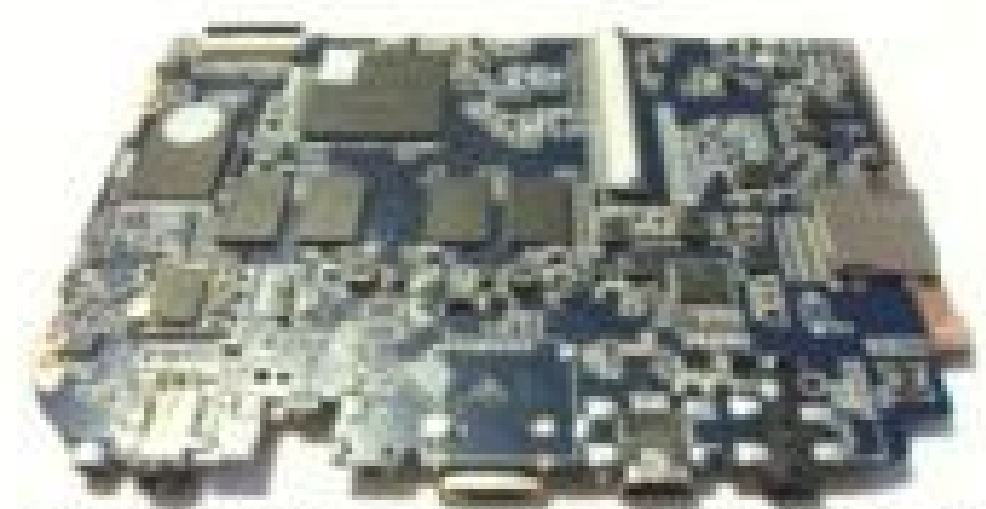
Und viele, viele mehr ...



PandaBoard



BeagleBoard



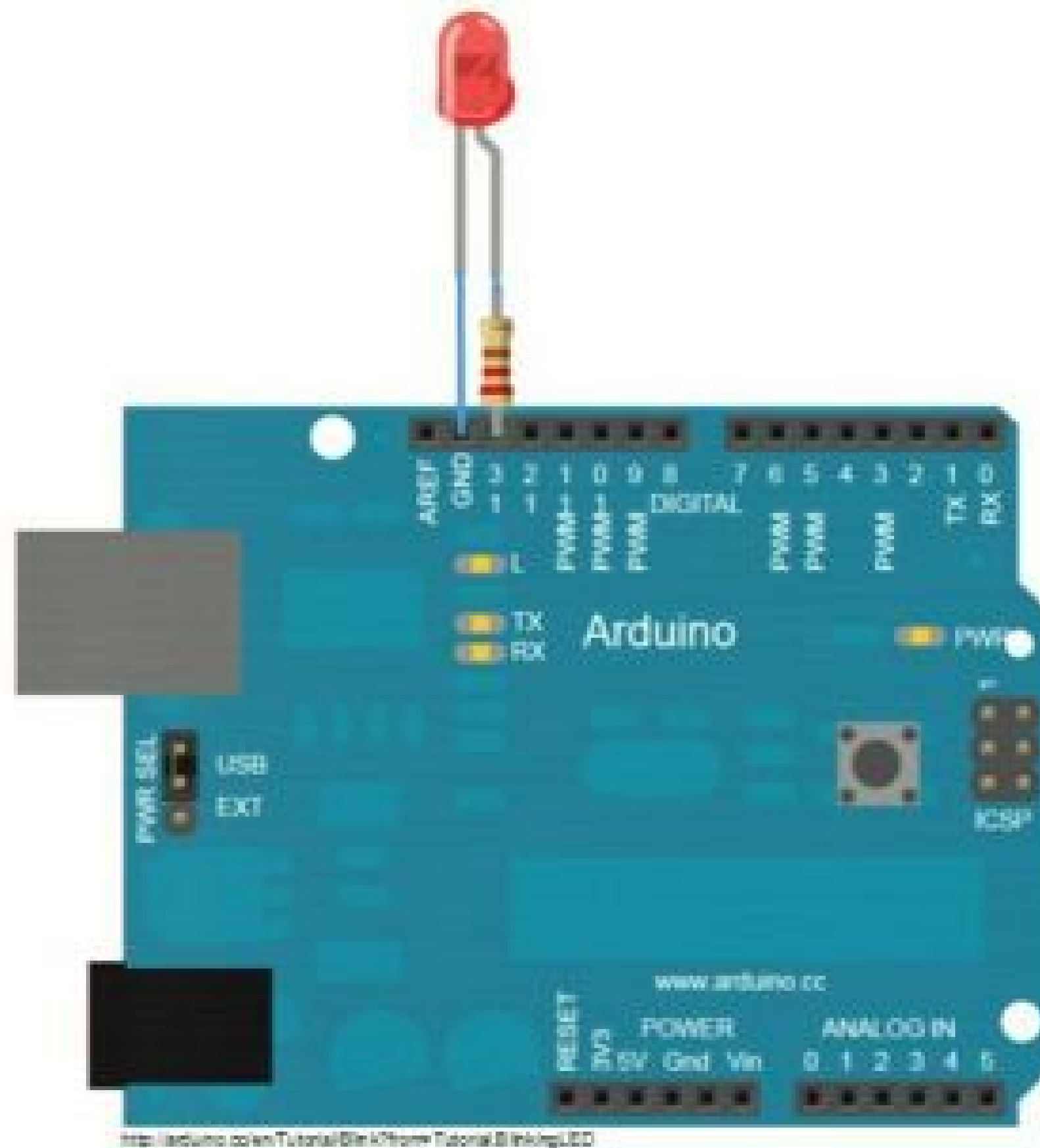
Gooseberry

Plattformen im direkten Vergleich



<http://realhackerspoint.blogspot.de/2013/04/raspberry-pi-vs-arduino-vs-beagleboard.html>

„Hallo Welt“ für Arduino





```
sketch_jun14a$
```

Upload abgeschlossen

Binäre Sketchgröße: 466 Bytes (von einem Maximum von 32.256 Bytes)





sketch_jun14a\$

```
void setup() {
```

```
}
```

Upload abgeschlossen

Binäre Sketchgröße: 466 Bytes (von einem Maximum von 32.256 Bytes)



„Hallo Welt“ für



<http://arduino.cc/en/Tutorial/Blink> www.tutorialspoint.com/Arduino/ArduinoLED

Einführung

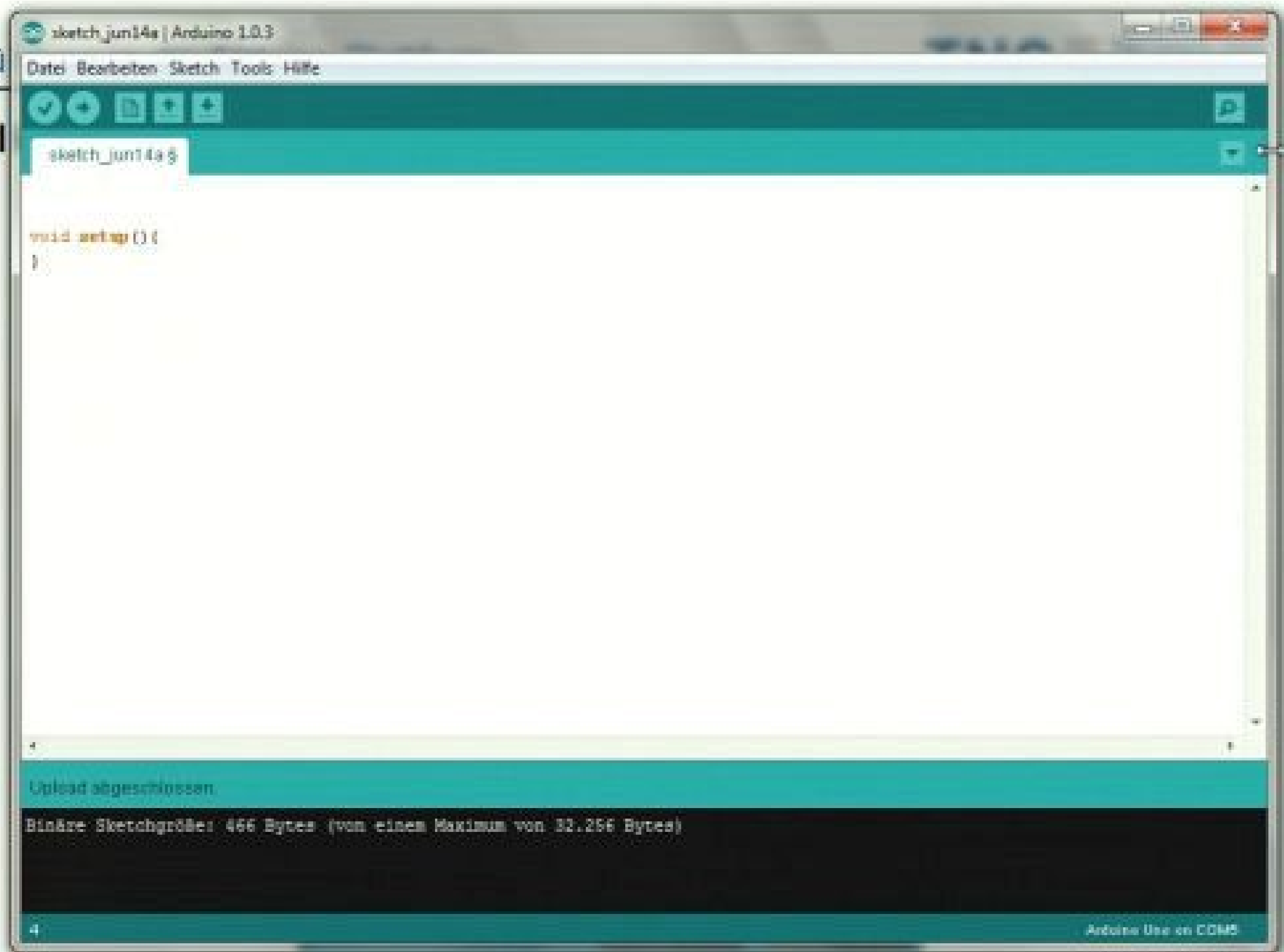
„Hallo“



<http://arduino.cc/en/Tutorial/DigitalPinWritingLED>

Einführung

„Hallo“

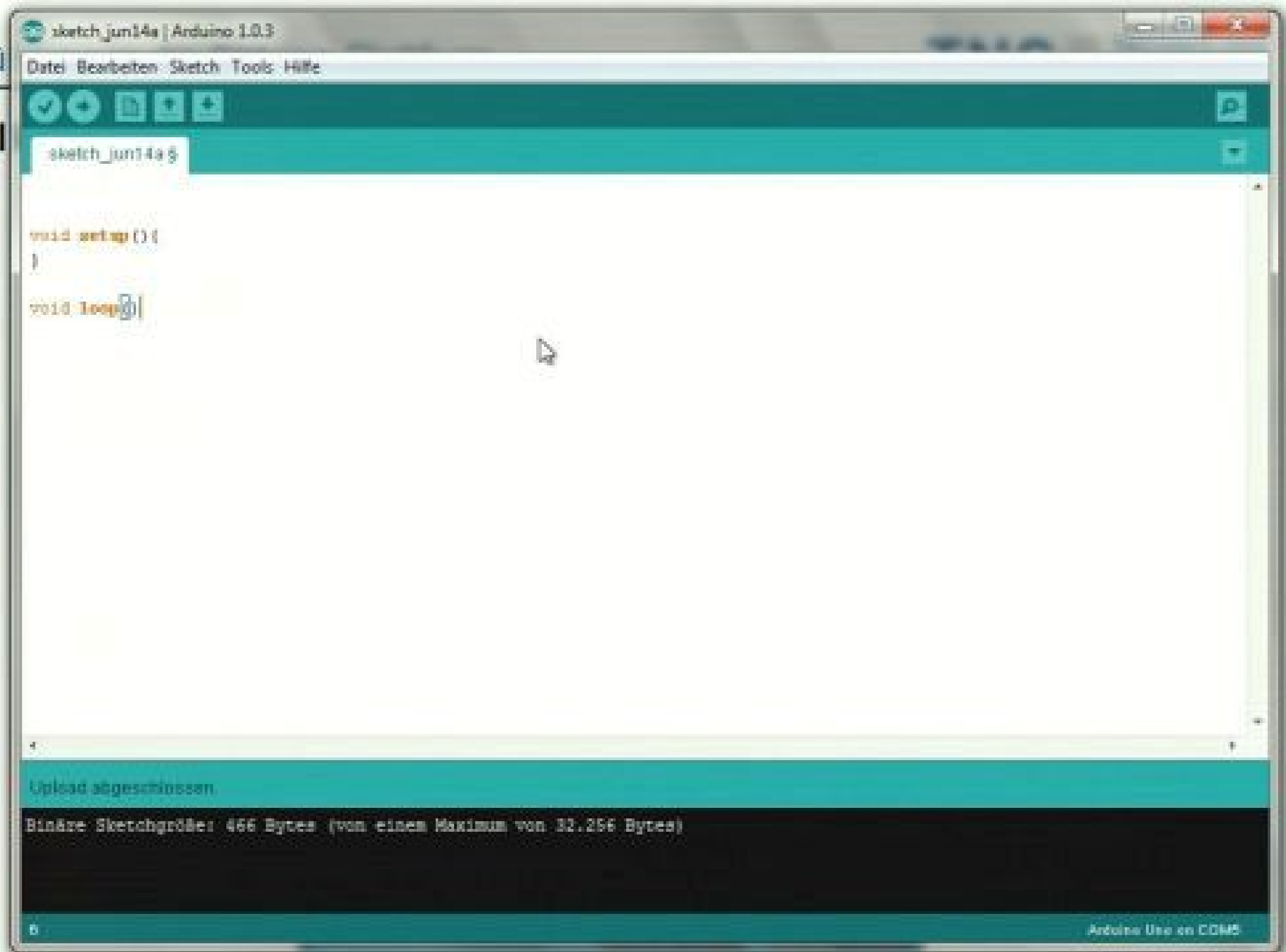


<http://arduino.cc/en/Tutorial/Blinky> <http://arduino.cc/en/Tutorial/BlinkingLED>



Einführung

„Hallo“

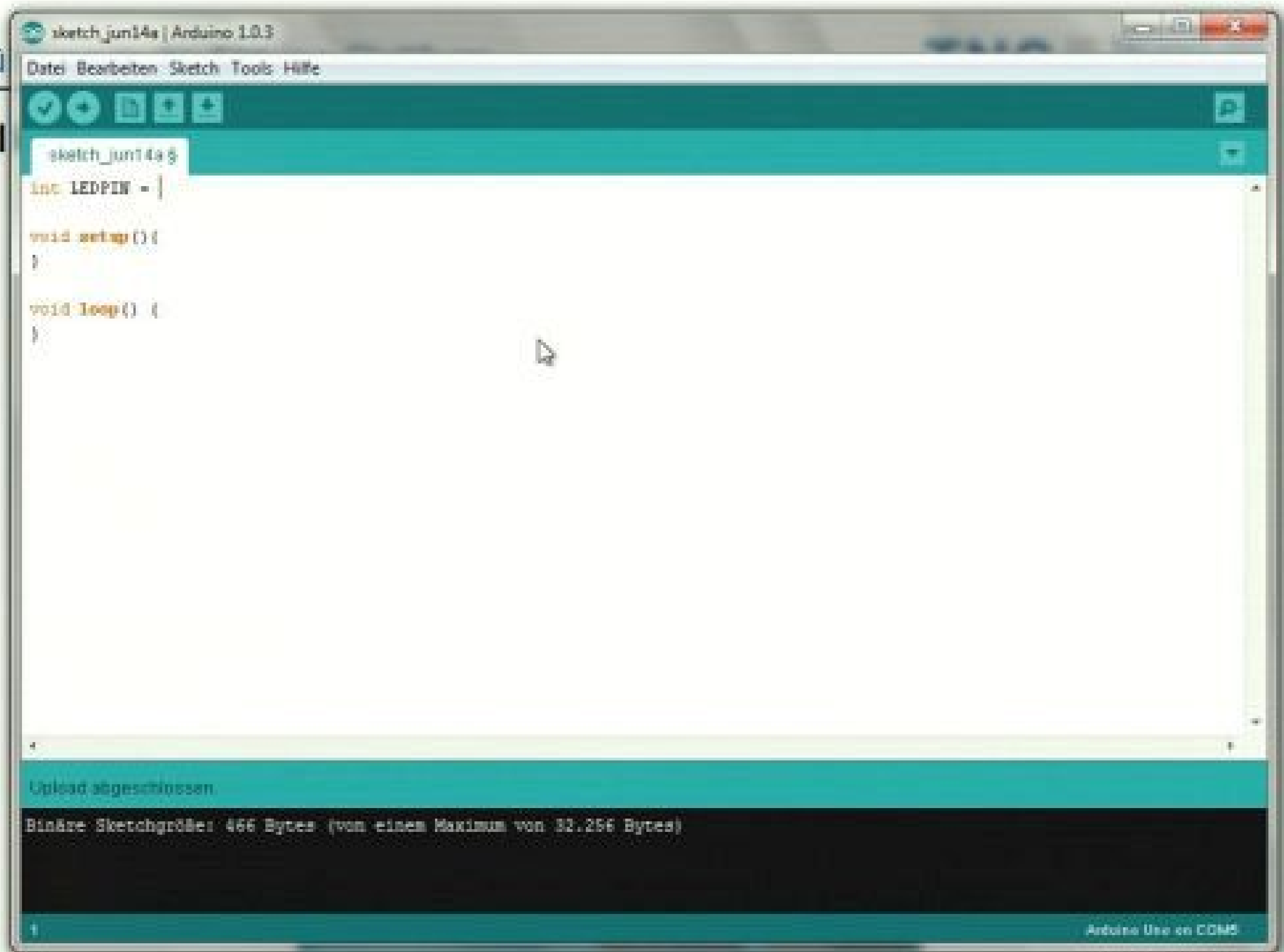


<http://arduino.cc/en/Tutorial/DigitalPinWritingLED>



Einführung

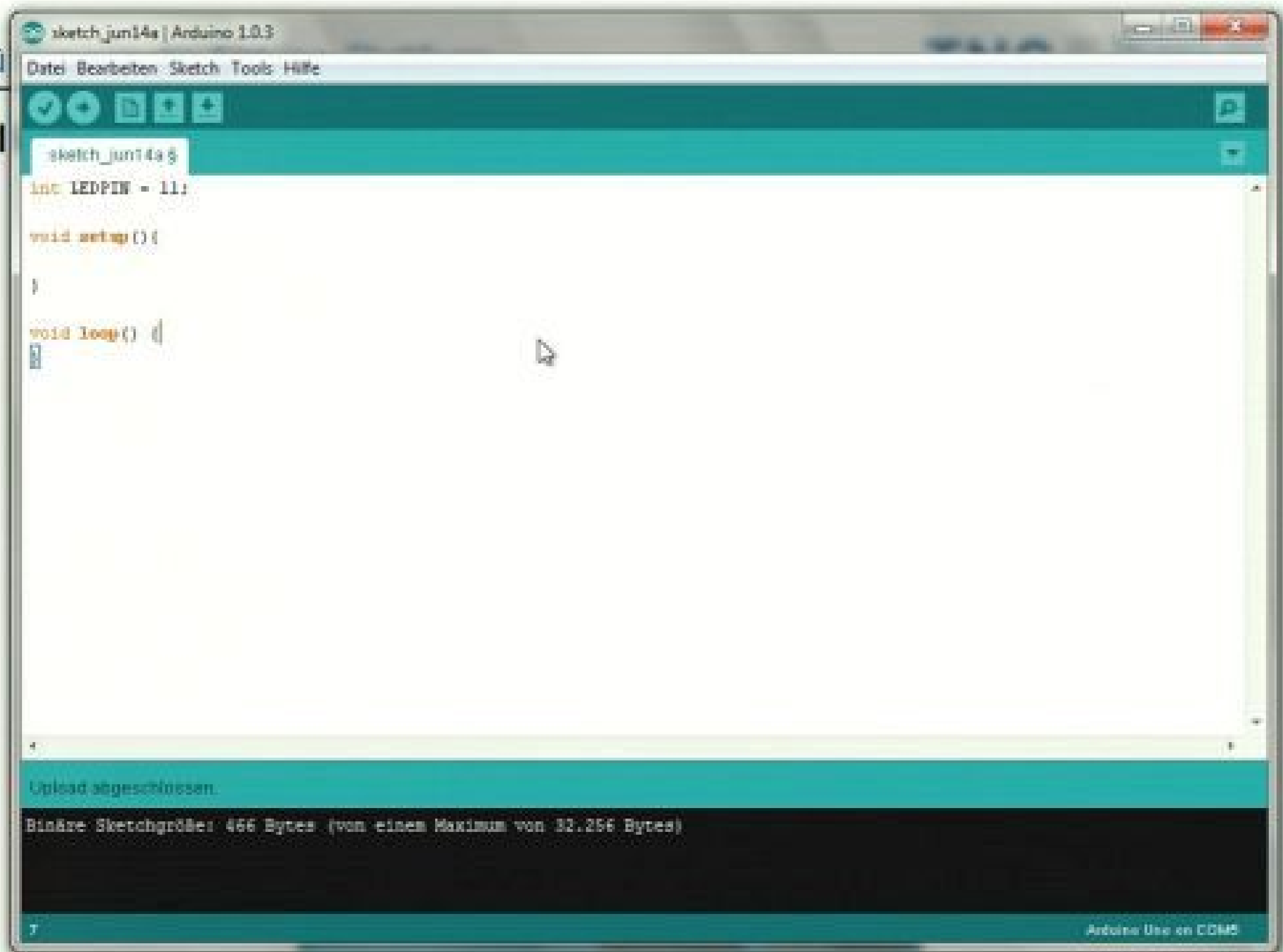
„Hallo“



<http://arduino.cc/en/Tutorial/BlinkingLED>



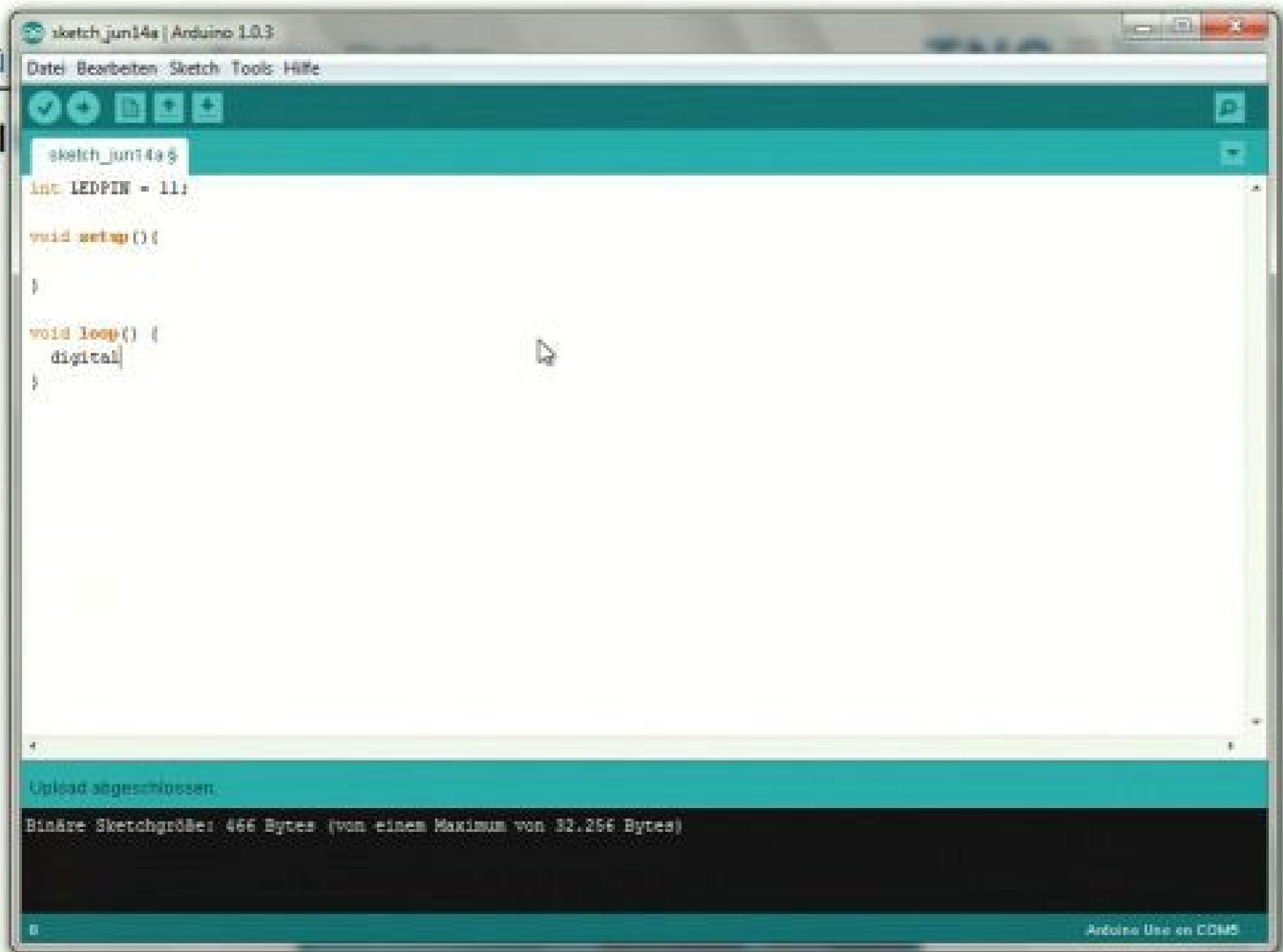
Einführung
„Hallo“



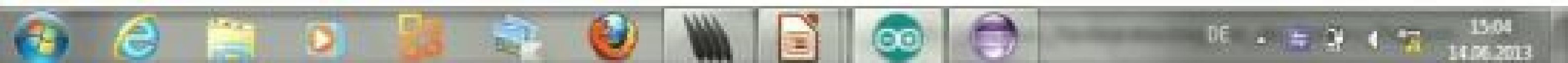
<http://arduino.cc/en/Tutorial/Blink> www.tutorialspoint.com/Arduino/Arduino_Blinking_LED



Einführung
„Hallo“

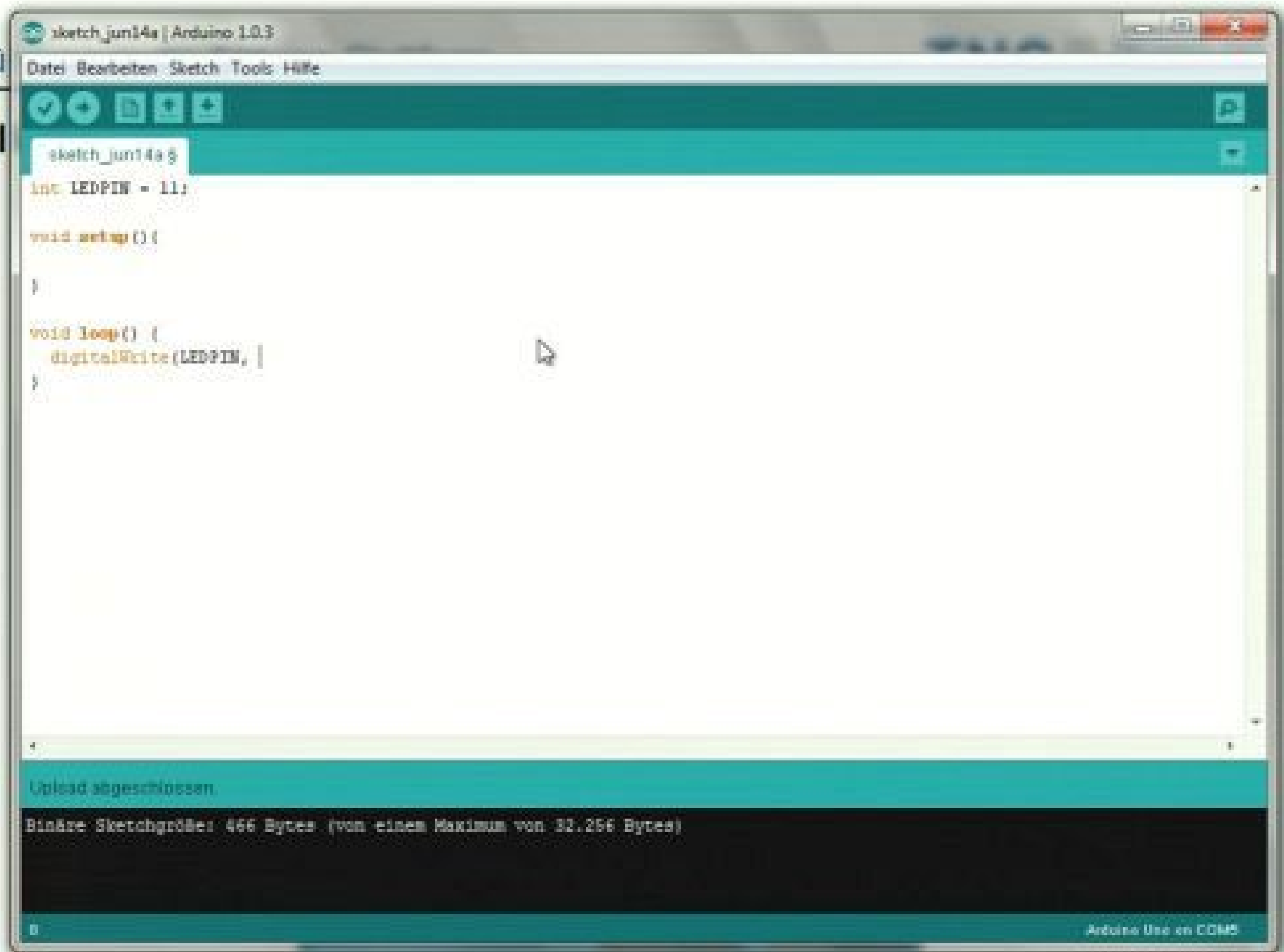


<http://arduino.cc/en/Tutorial/Blink> [Arduino Tutorial: Blinking LED](#)



Einführung

„Hallo“

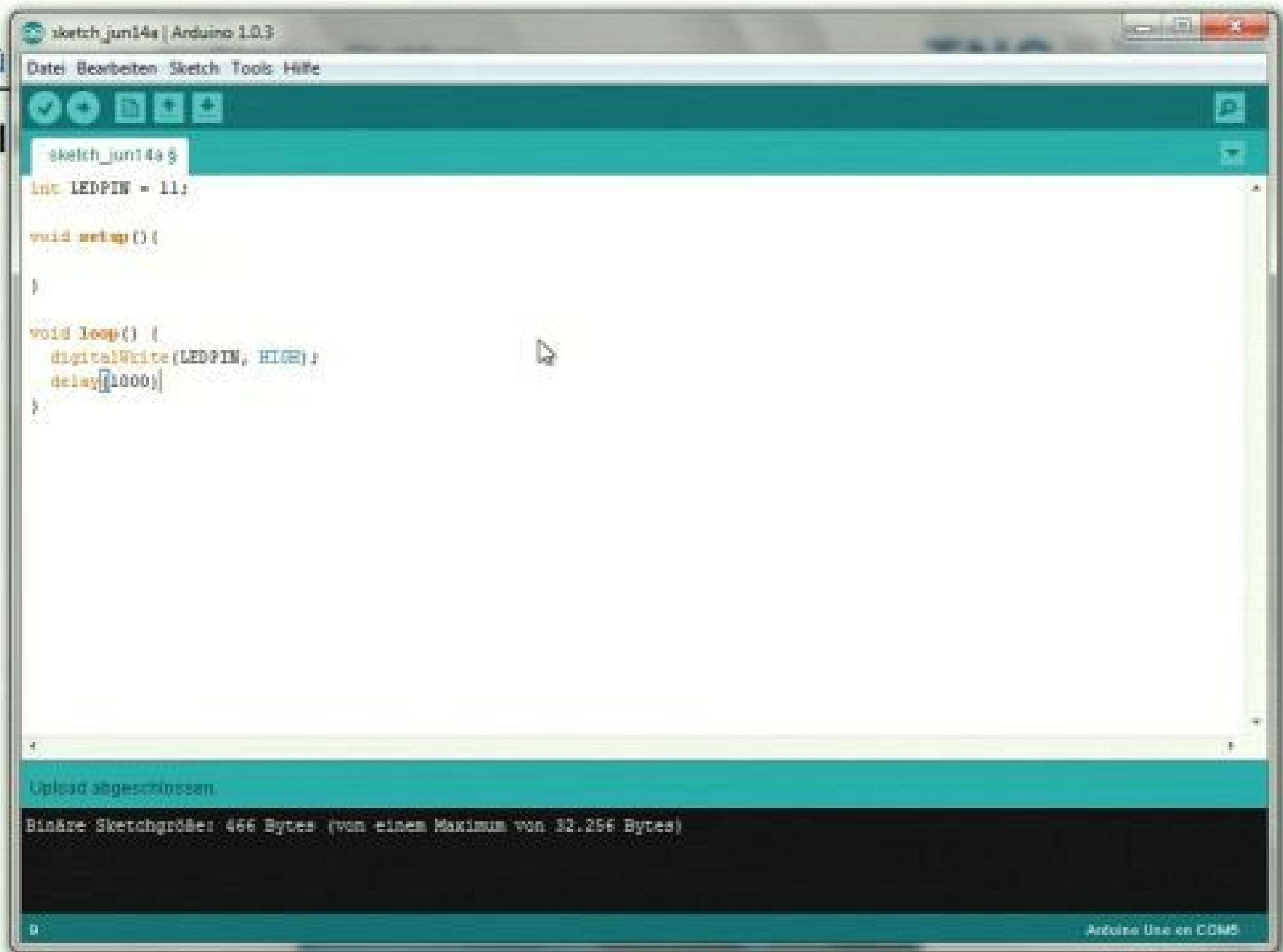


<http://arduino.cc/en/Tutorial/BlinkingLED>



Einführung

„Hallo“

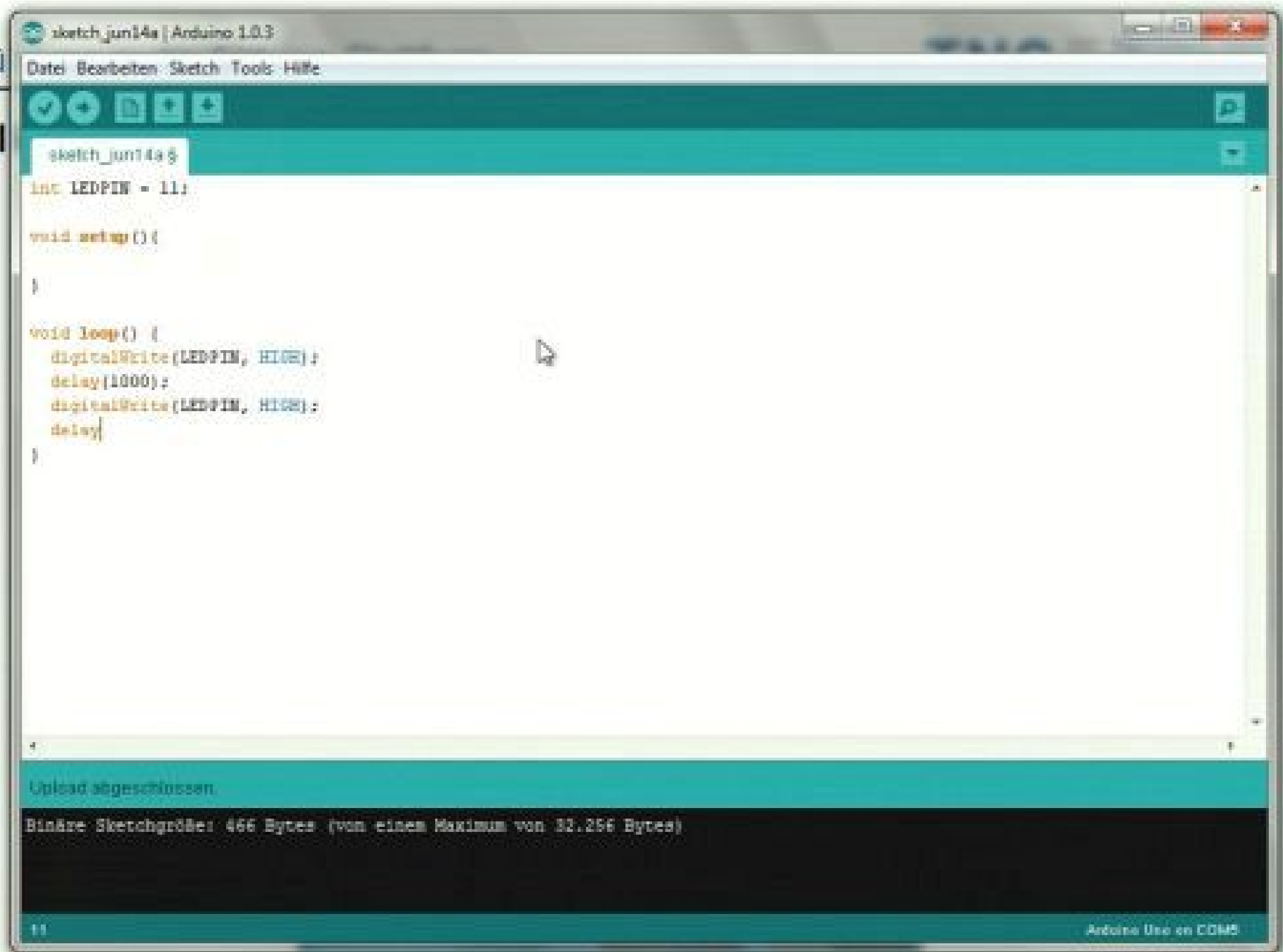


<http://secondcode.de/Tutorials/Arduino-Tutorial-BlinkingLED>

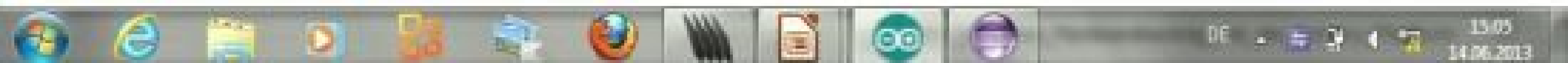


Einführung

„Hallo“

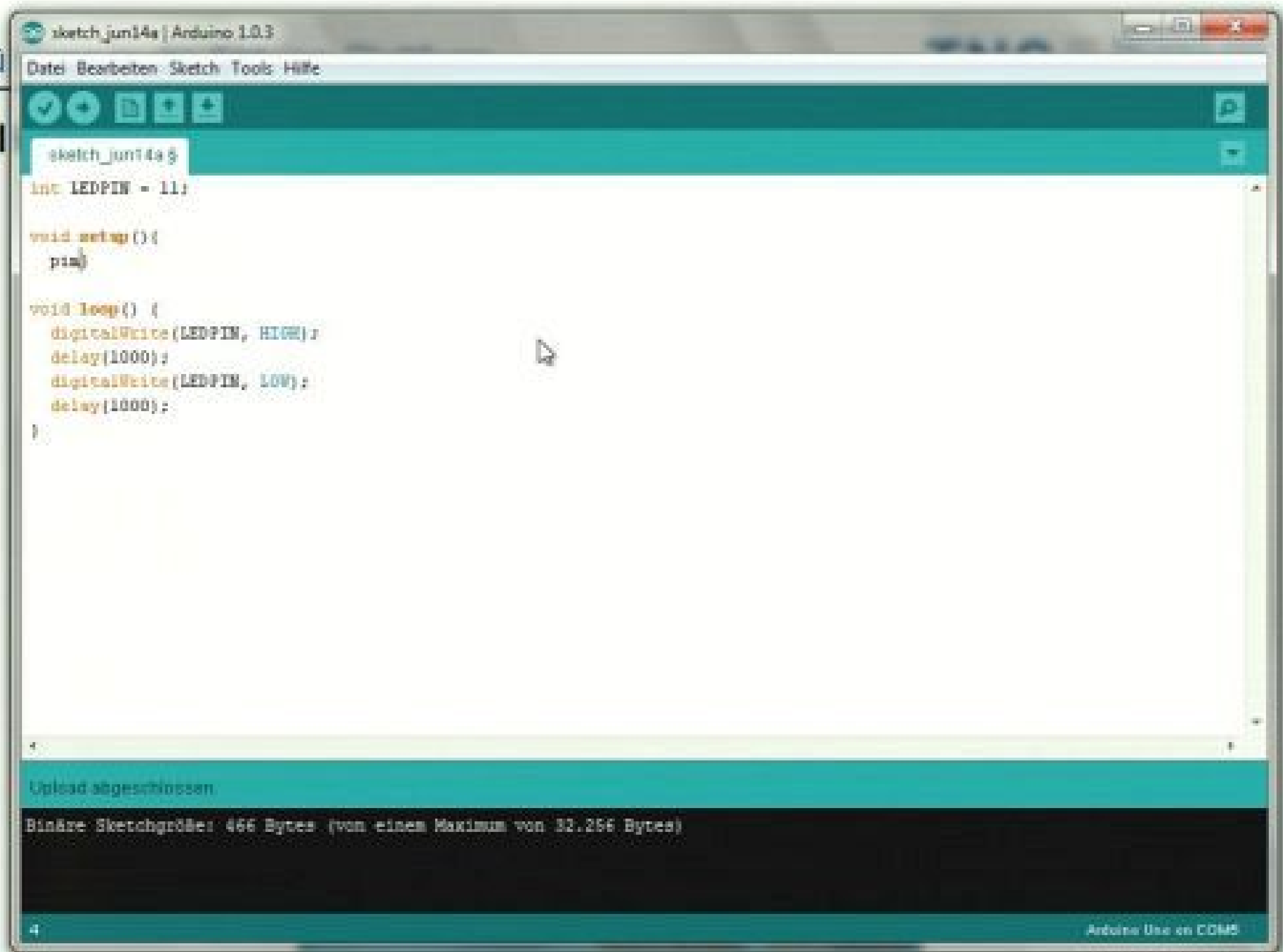


<http://arduino.cc/en/Tutorial/DigitalPinWritingLED>



Einführung

„Hallo“

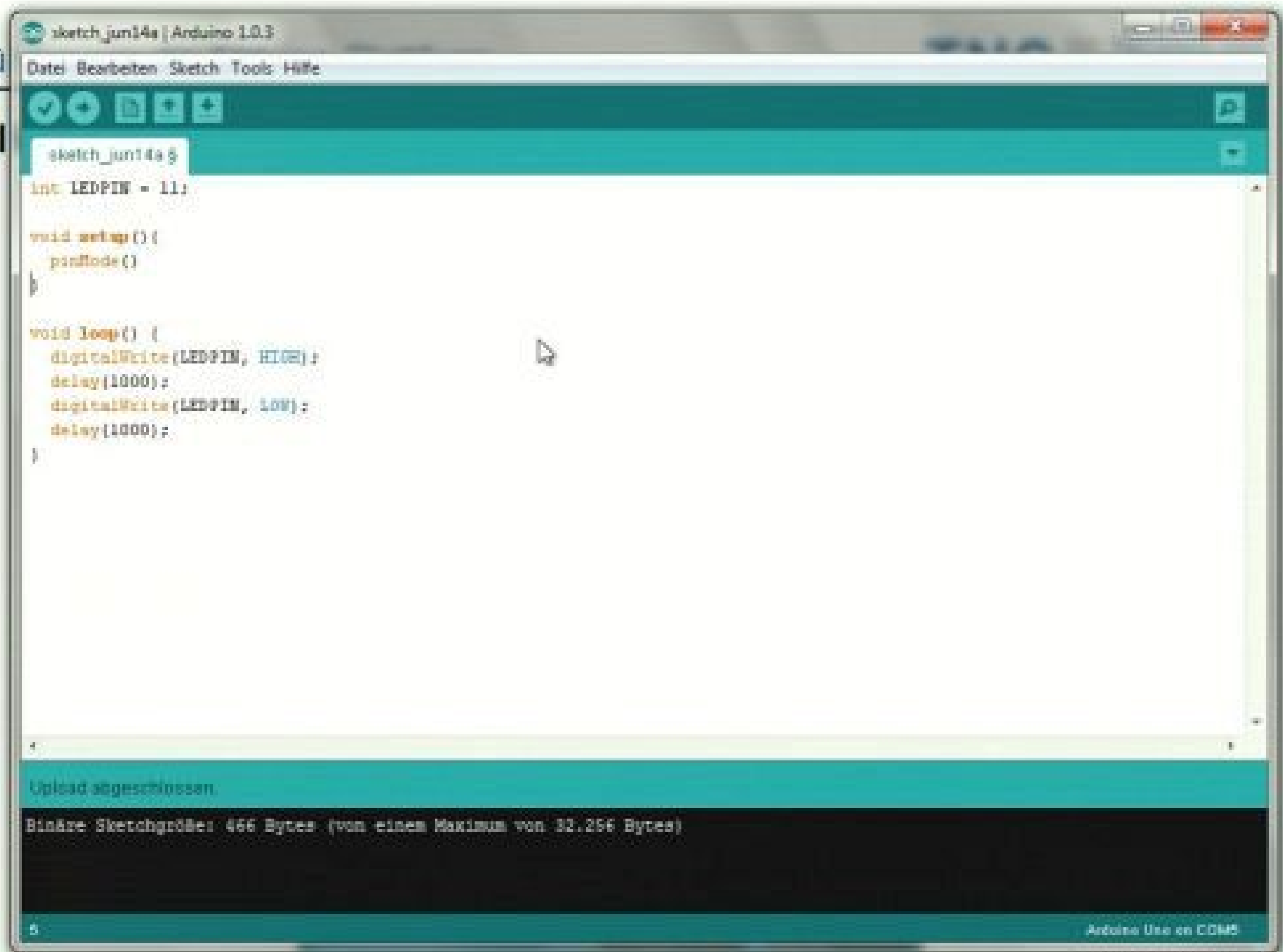


<http://arduino.cc/en/Tutorial/DigitalPinBlinkingLED>



Einführung

„Hallo“

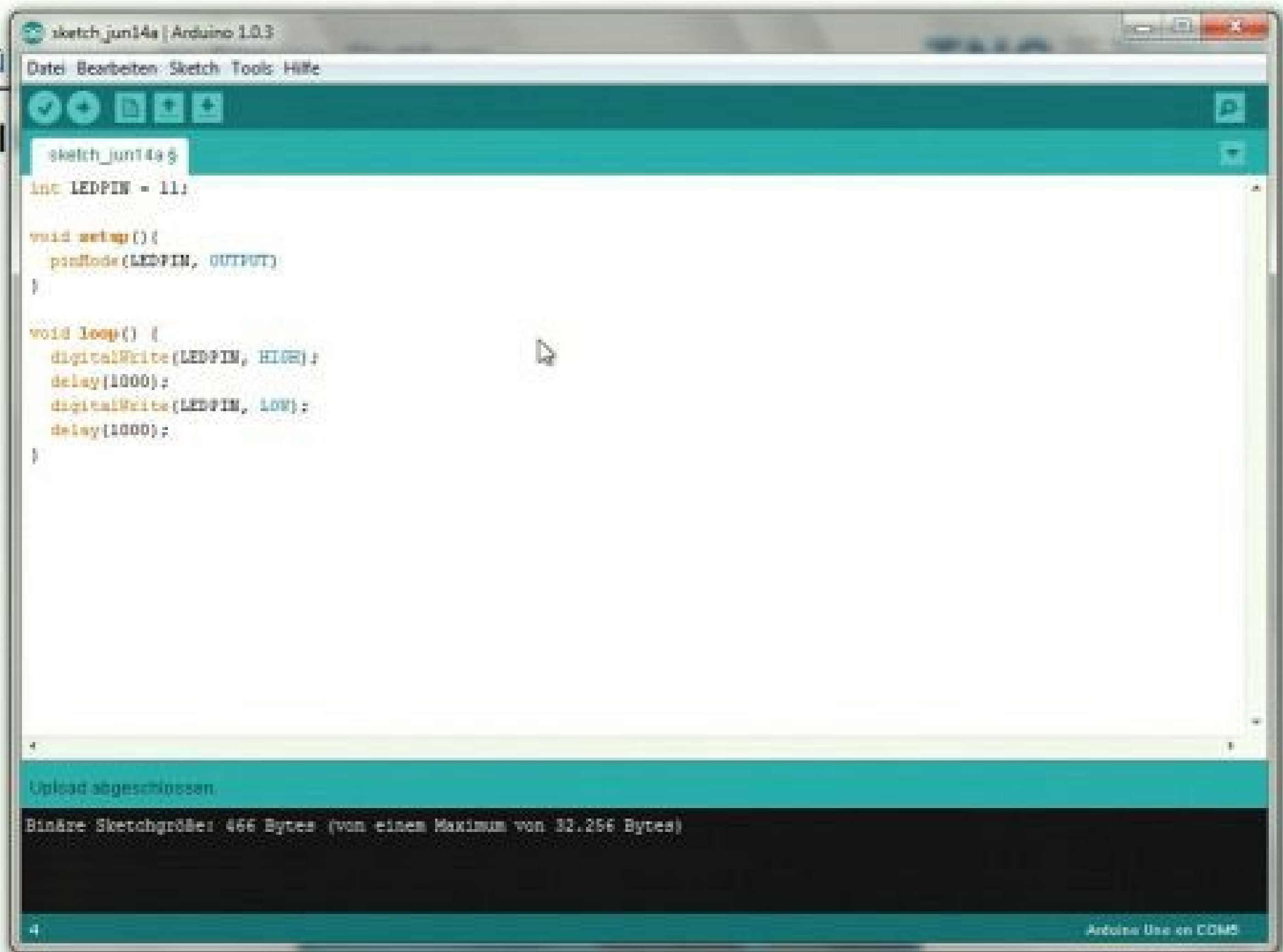


<http://arduino.cc/en/Tutorial/DigitalBlinkingLED>



Einführung

„Hallo“

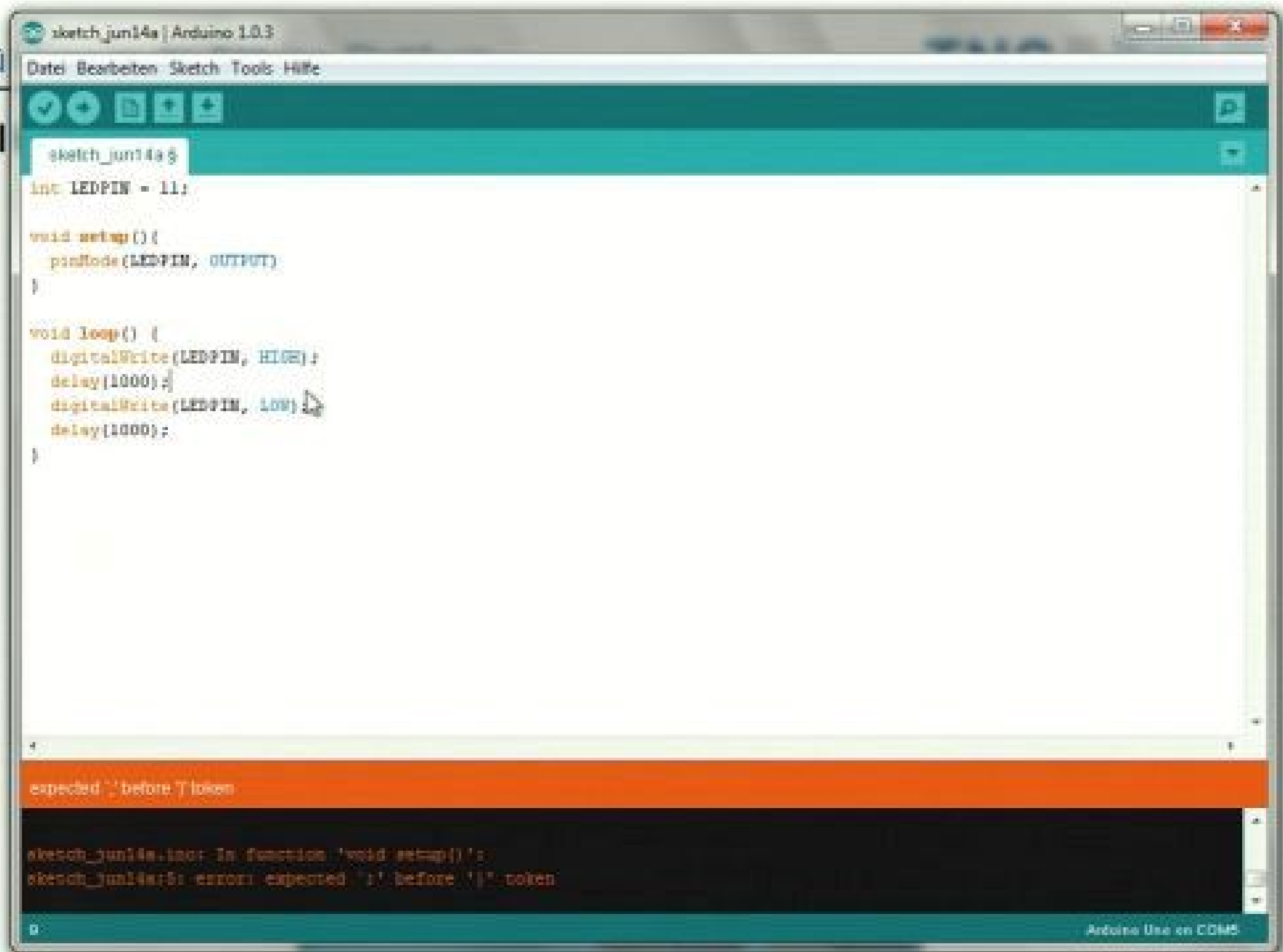


<http://arduino.cc/en/Tutorial/Blink> [Arduino Tutorial: Blinking LED](#)



Einführung

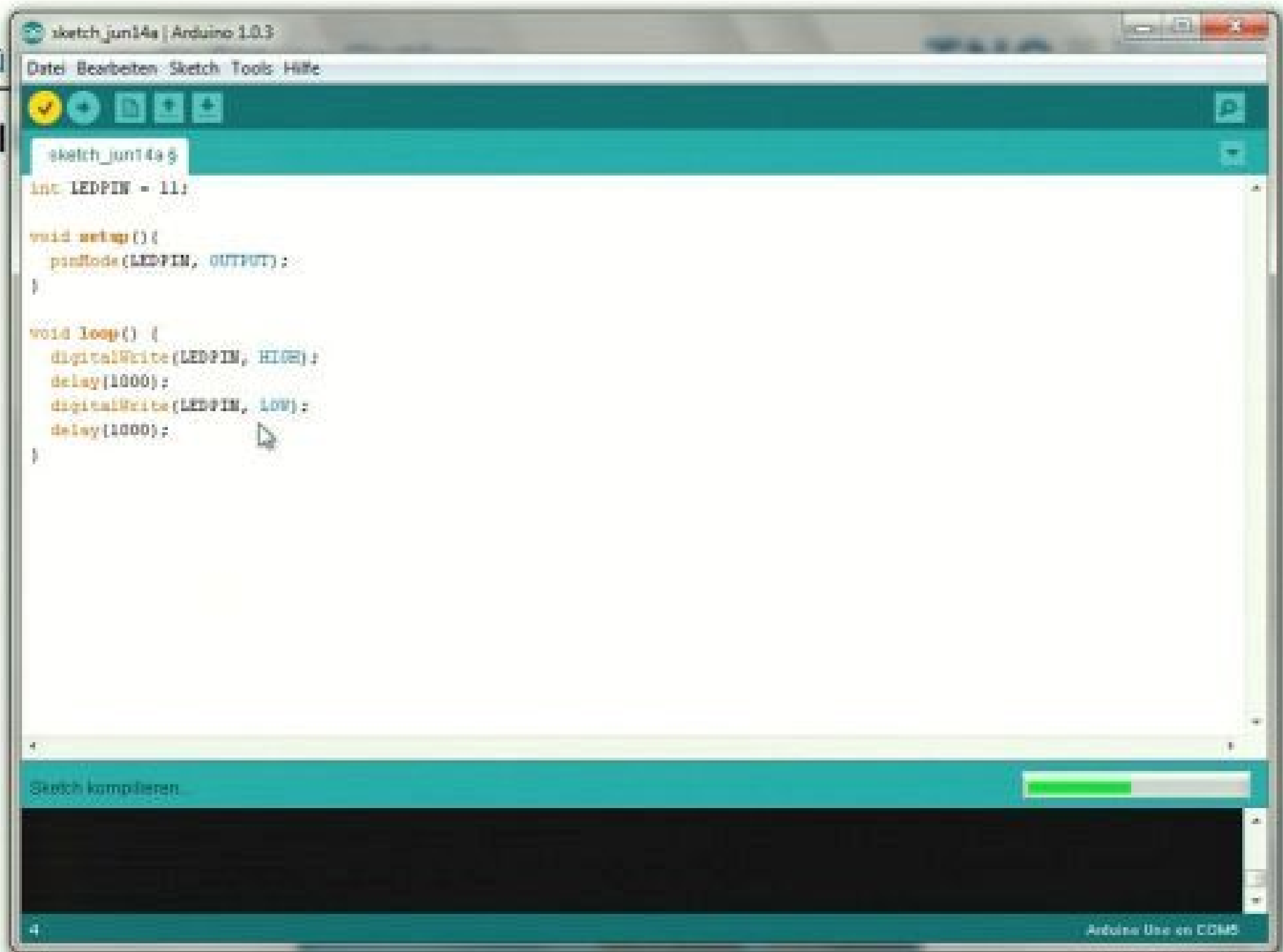
„Hallo“



<http://arduino.cc/en/Tutorial/BlinkingLED>

Einfo

„Hal

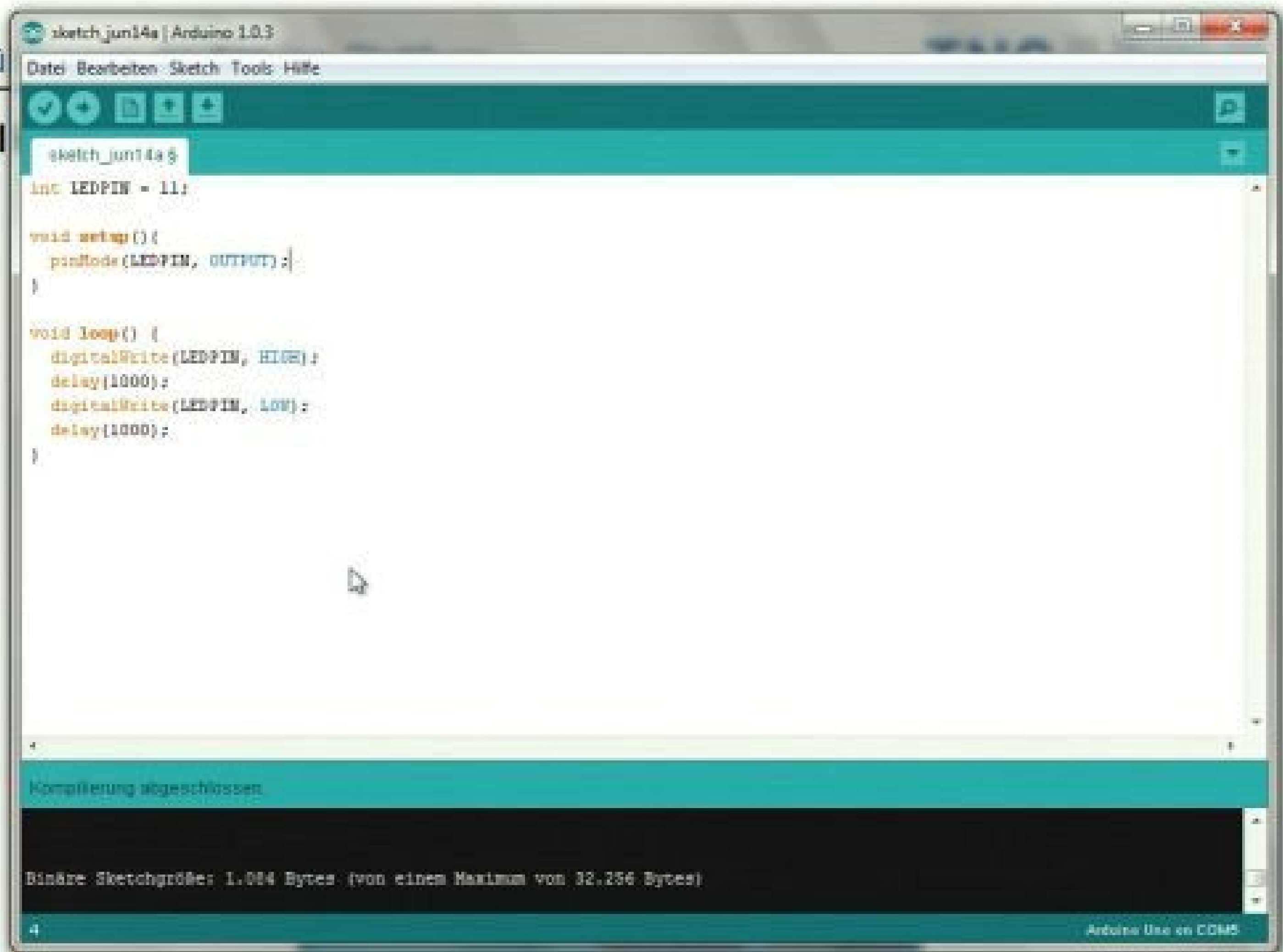


<http://secondcorenTutorialDieKleineTutorialBlinkingLED>



Einführung

„Hallo“

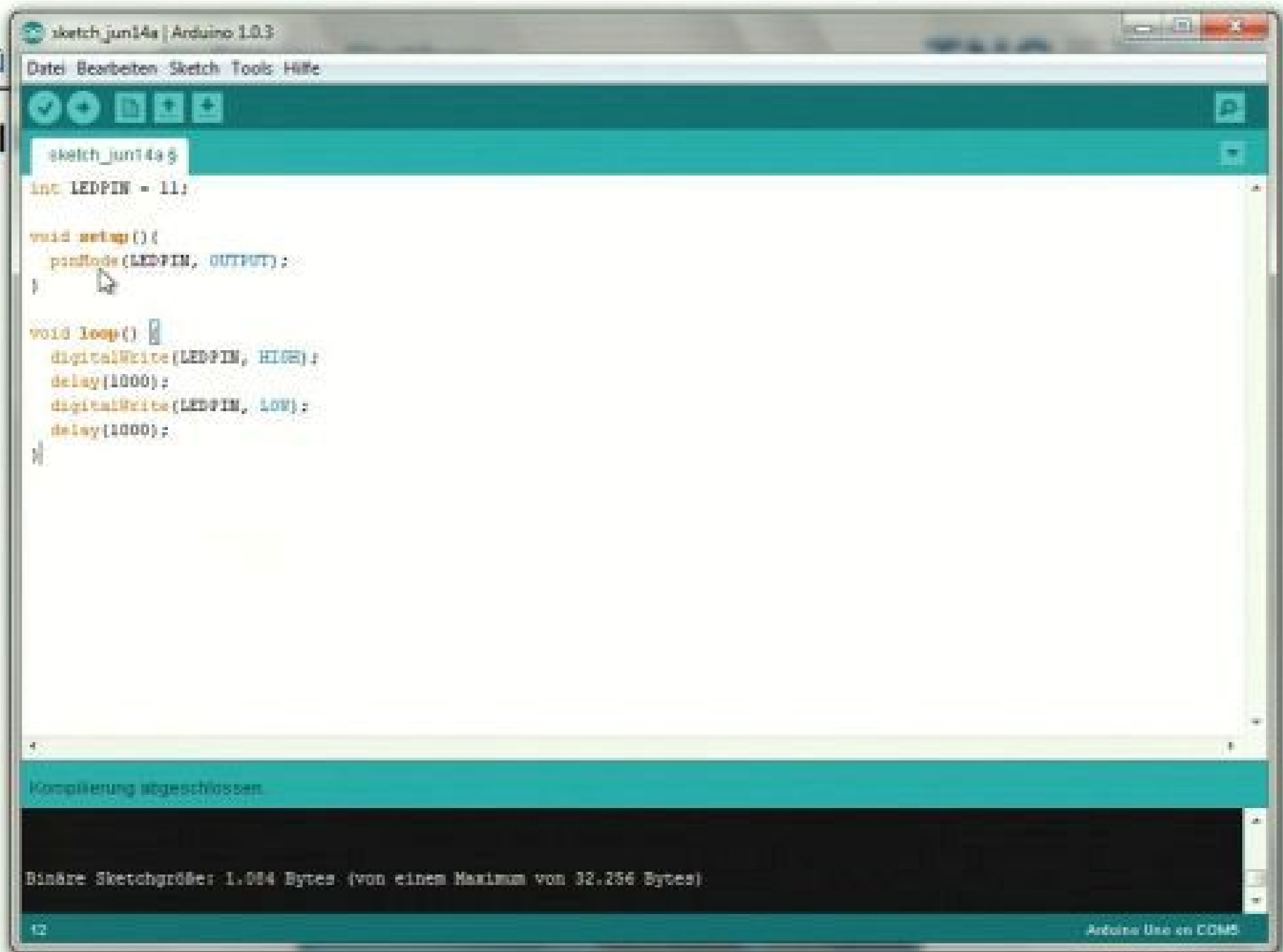


<http://arduino.cc/en/Tutorial/BlinkingLED>



Einführung

„Hallo“

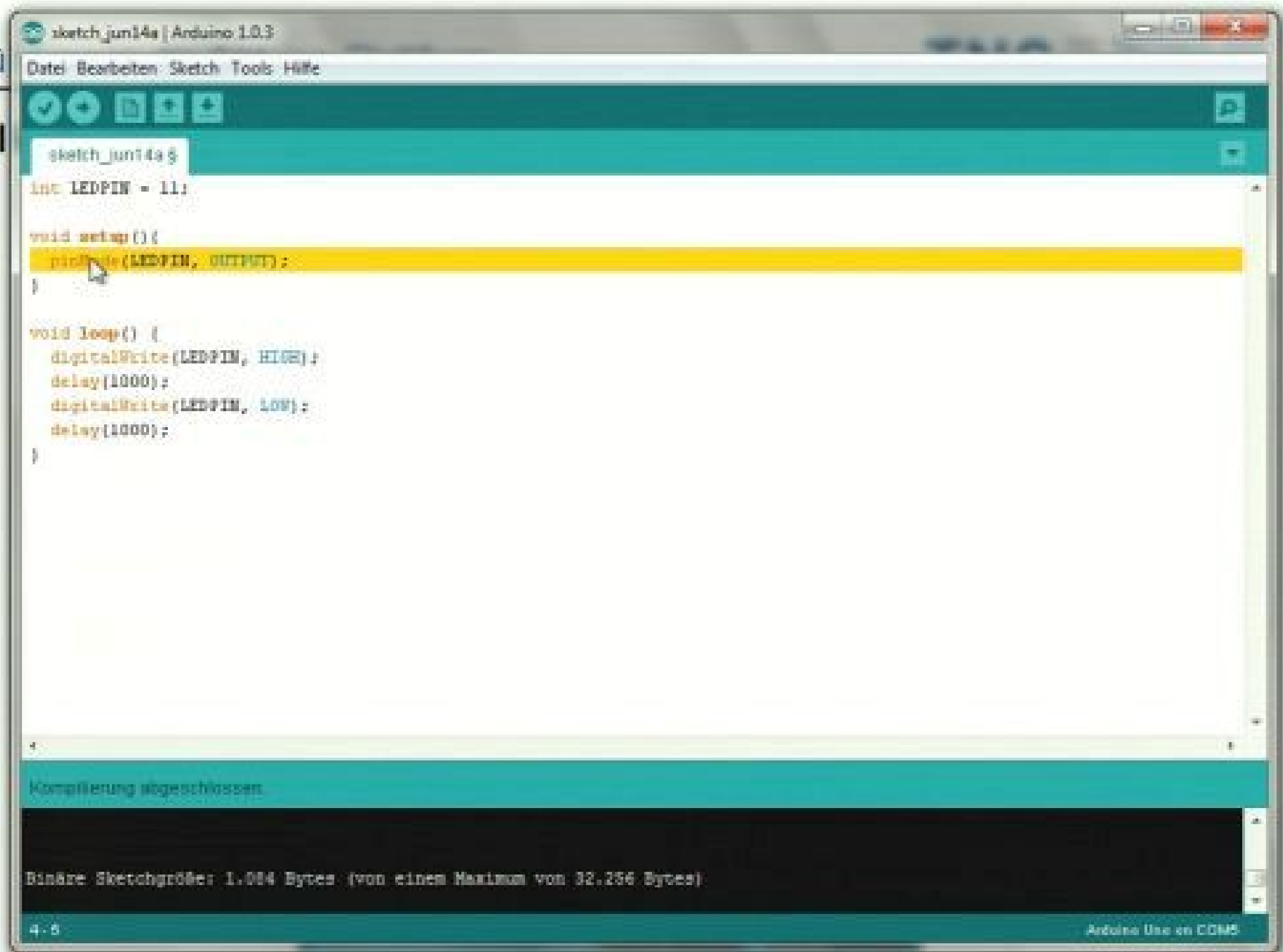


<http://arduino.cc/en/Tutorial/Blink> Die Arduino-Tutorial-BlinkingLED



Einführung

„Hallo“

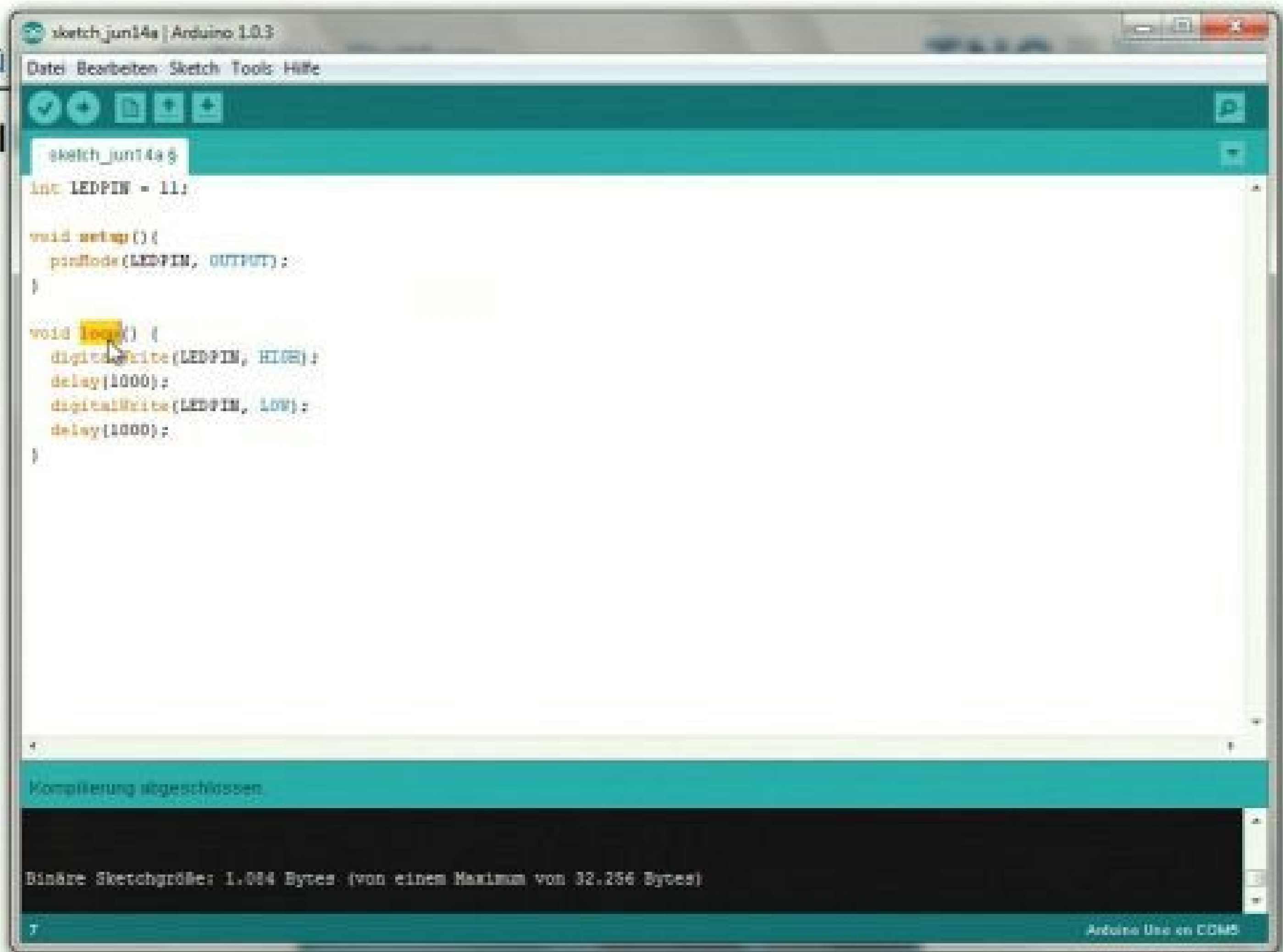


<http://arduino.cc/en/Tutorial/BlinkingLED>



Einführung

„Hallo“

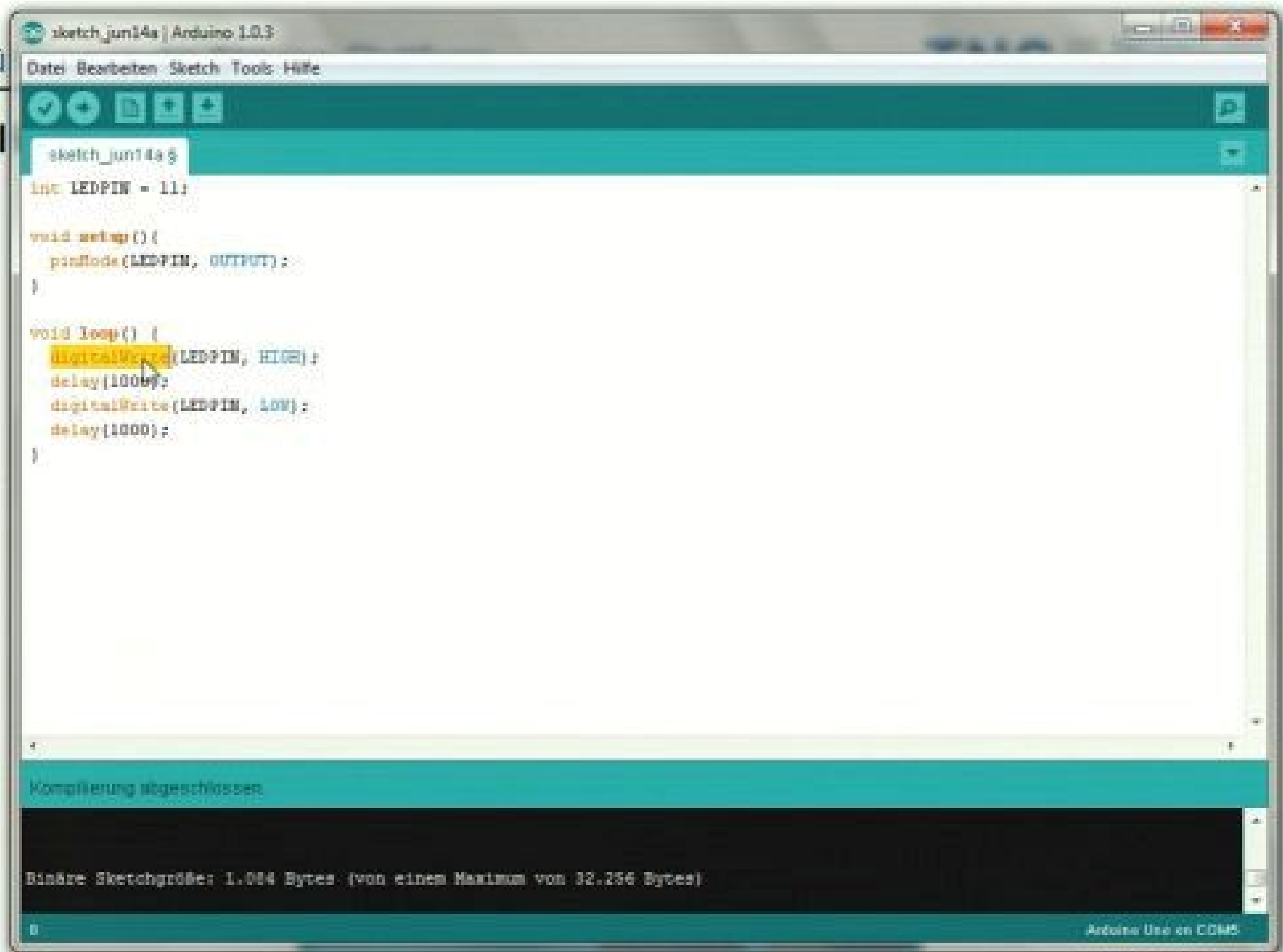


<http://secondcode.de/Tutorials/Arduino-Tutorial-BlinkingLED>



Einführung

„Hallo“

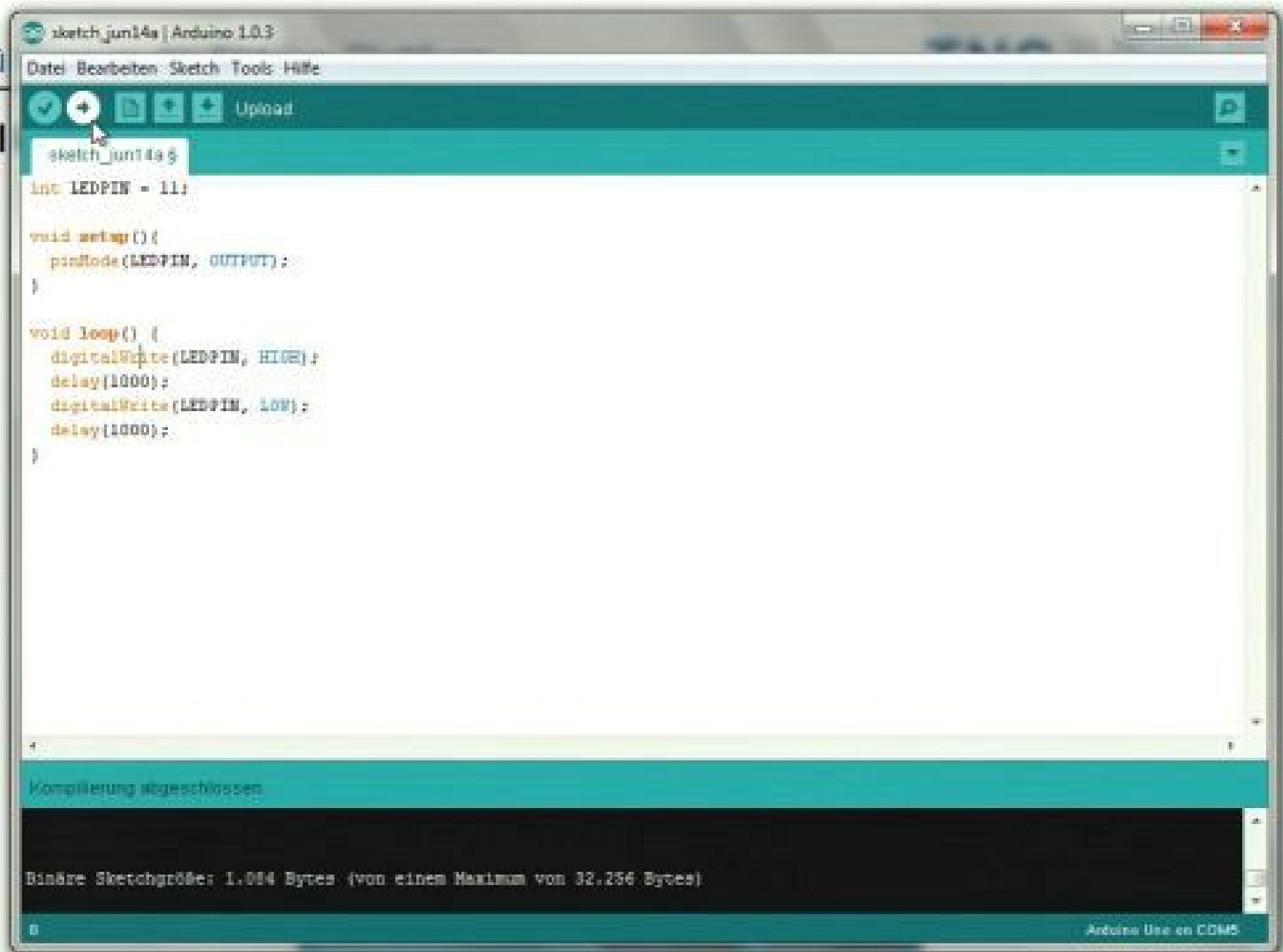


<http://arduino.cc/en/Tutorial/Blink> www.kitrow.com/tutorial/BlinkingLED



Einführung

„Hallo“

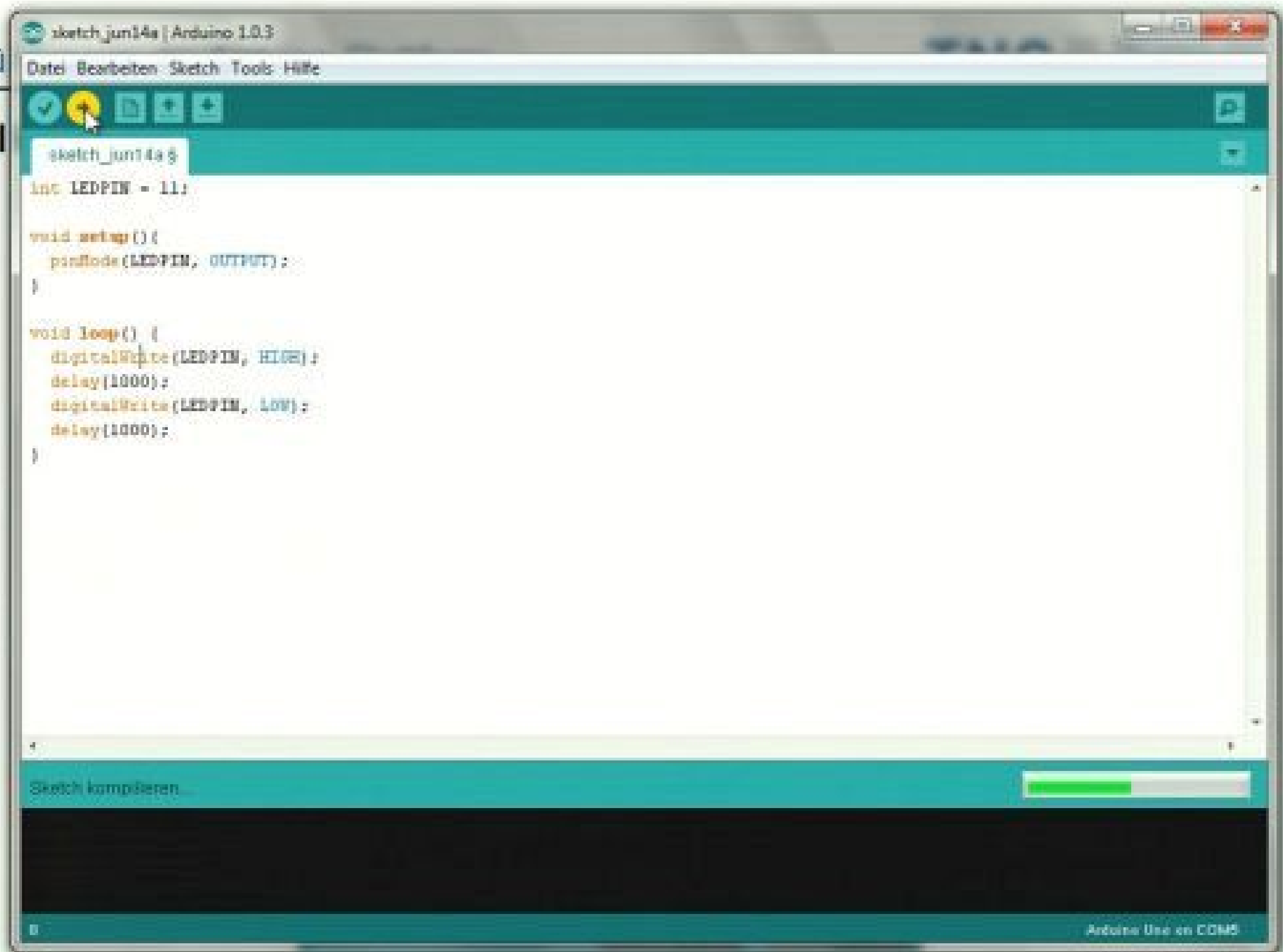


<http://arduino.cc/en/Tutorial/BlinkingLED>



Einführung

„Hallo“

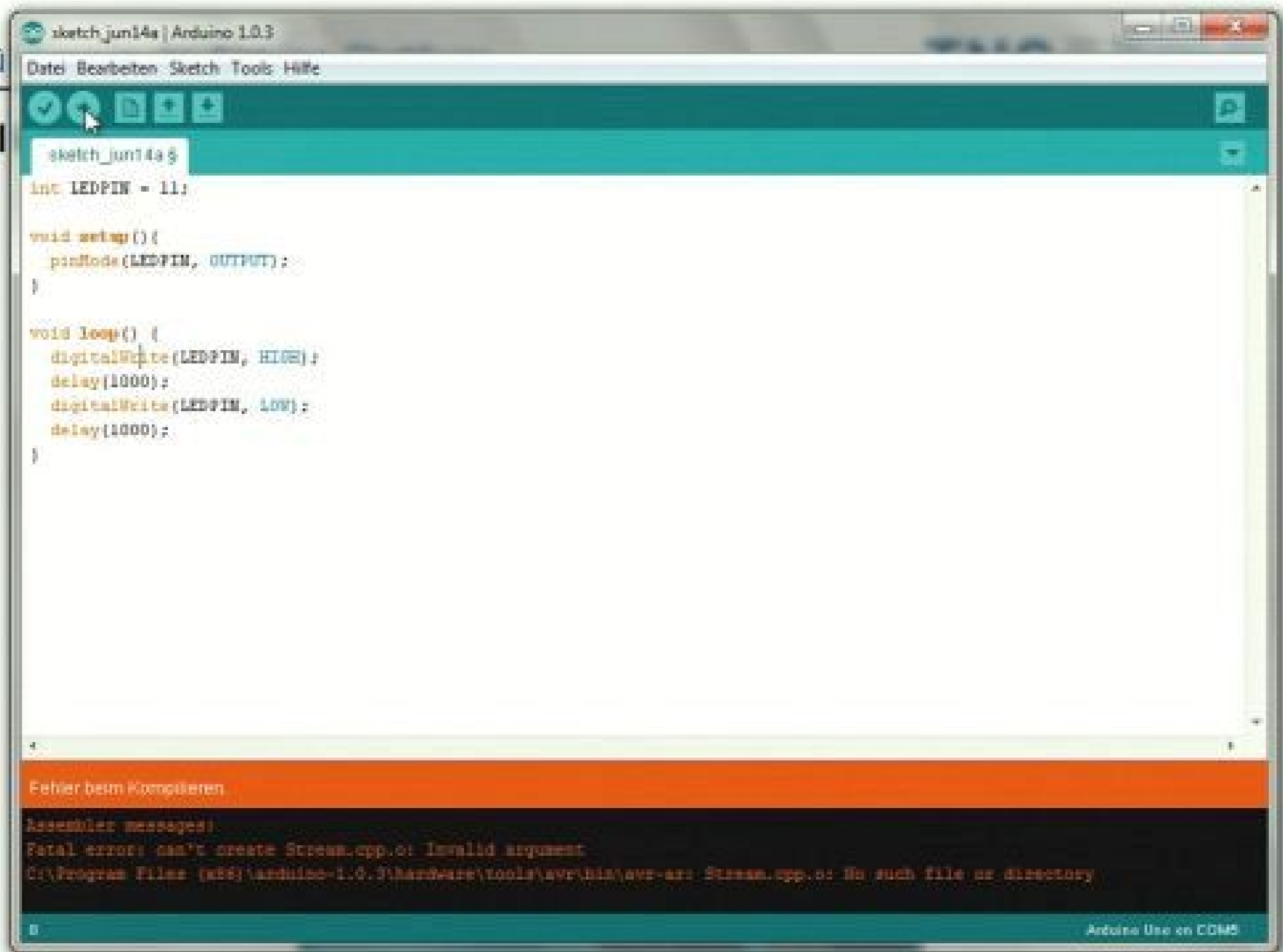


<http://secondcolumnTutorialBlog.com/Tutorial/blinkLED>



Einführung

„Hallo“

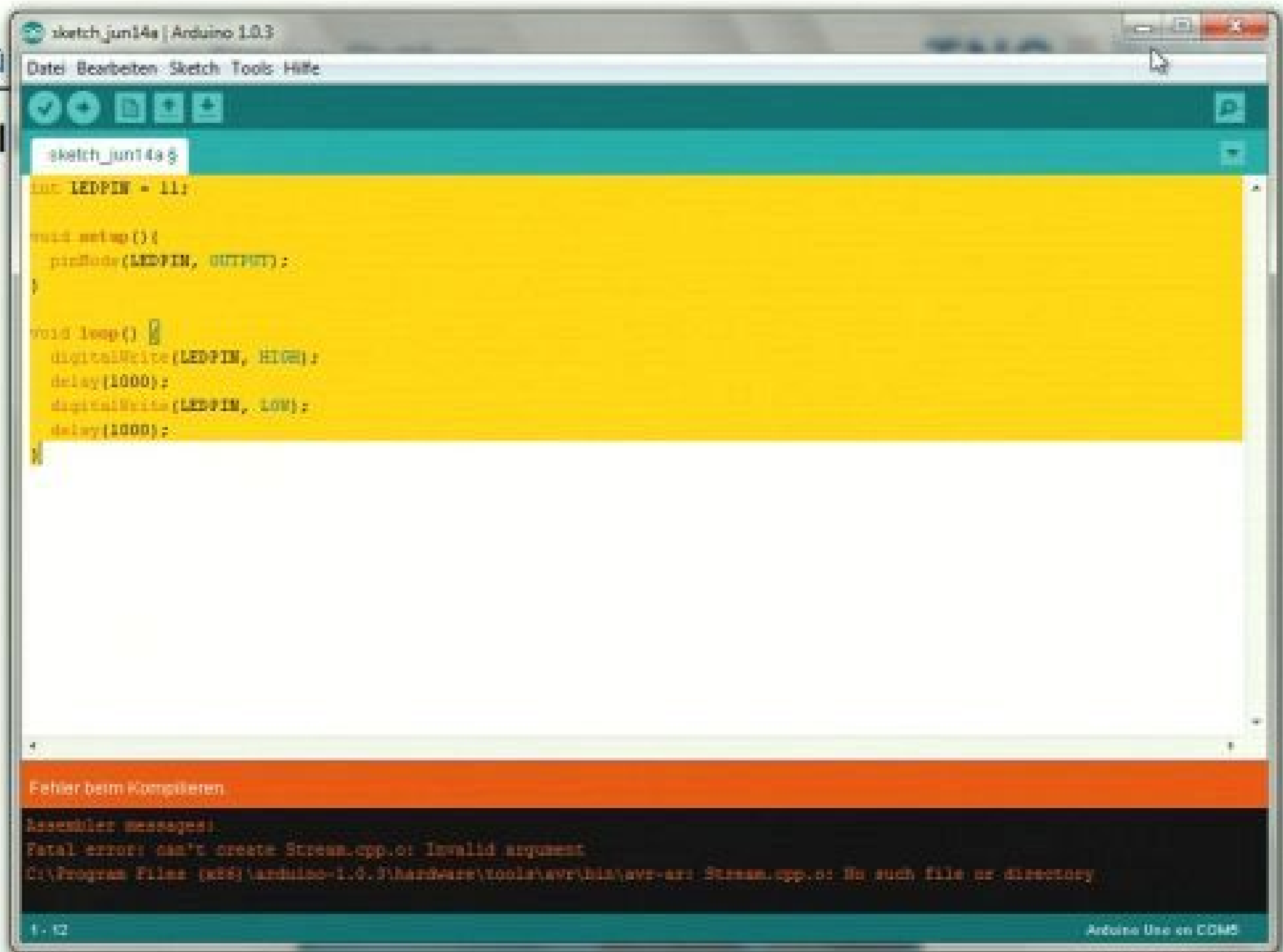


<http://secondcolumn.tutorialsinthekitchen.com/tutorial-5-blinking-led/>



Einführung

„Hallo“

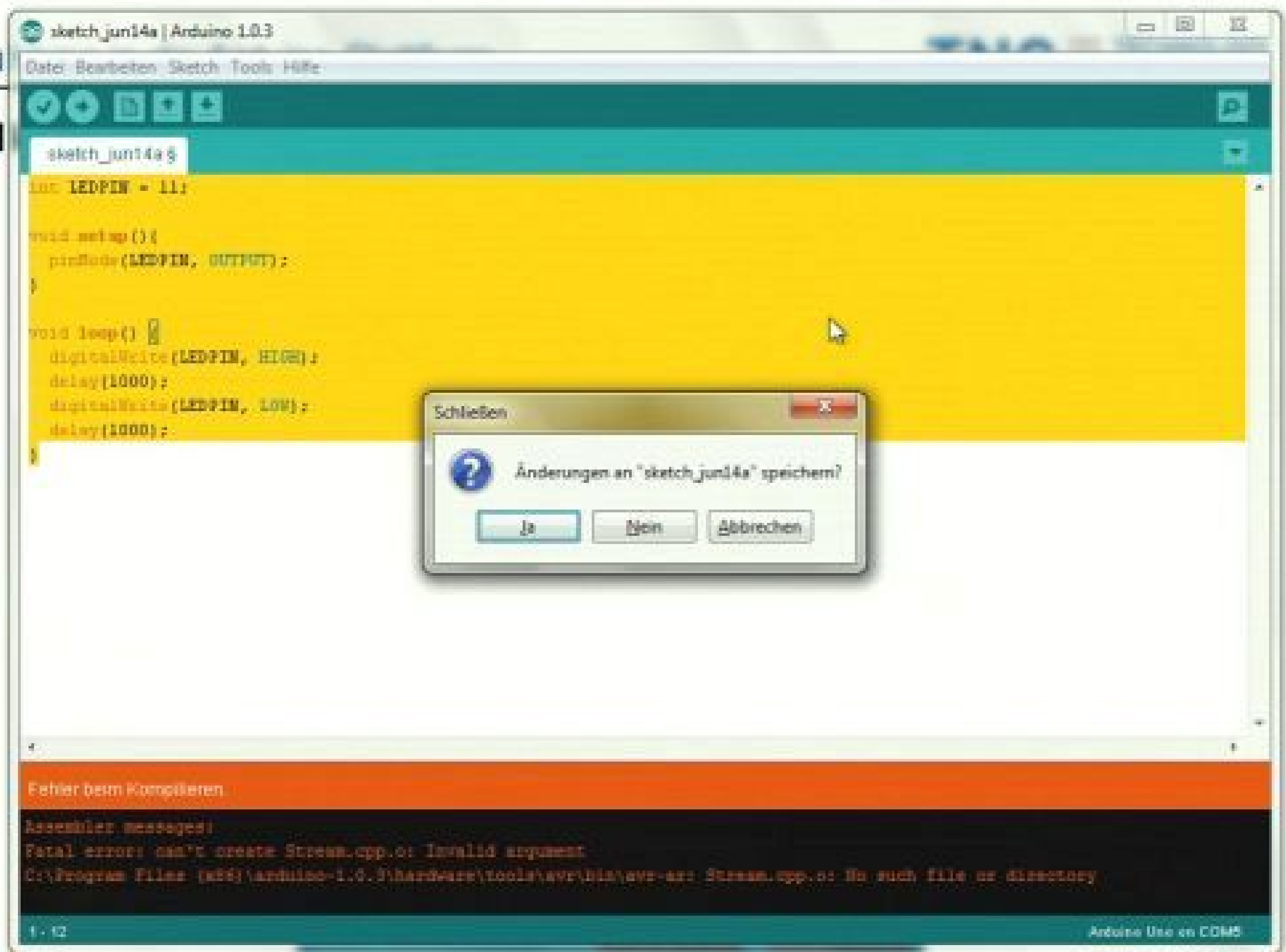


<http://secondcolumn.com/Tutorials/51e-Kit/Arduino-Tutorial-51e-Kit-LED>



Einführung

„Hallo“



<http://arduino.cc/en/Tutorial/BlitzKontrollTutoriel3IntegLED>



„Hallo Welt“ für

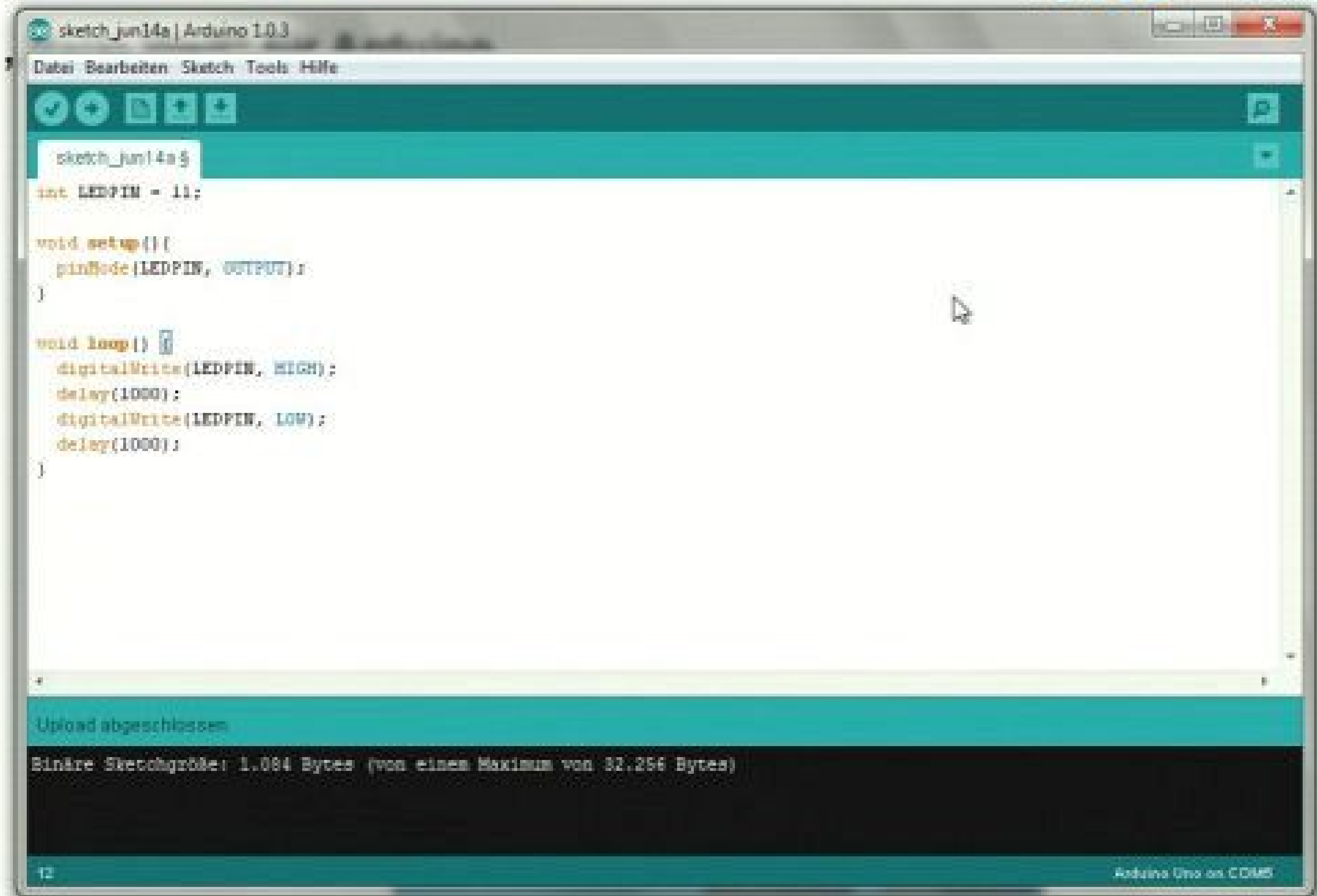


<http://arduino.cc/en/Tutorial/BlinkingLED>

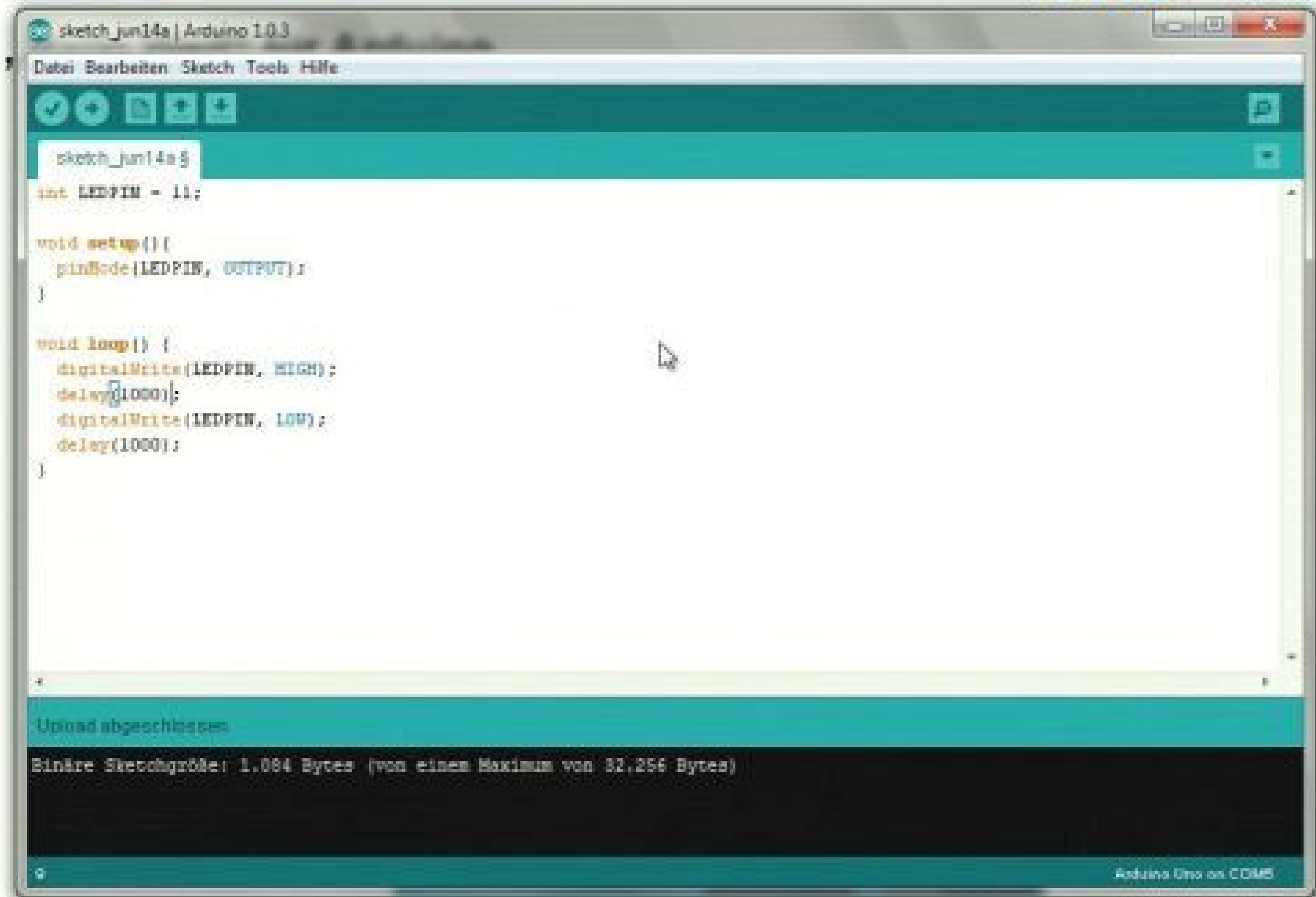
„Hallo Welt“ für



<http://arduino.cc/en/Tutorial/Blink> <http://www.tutorialspoint.com/Arduino/ArduinoLED.html>

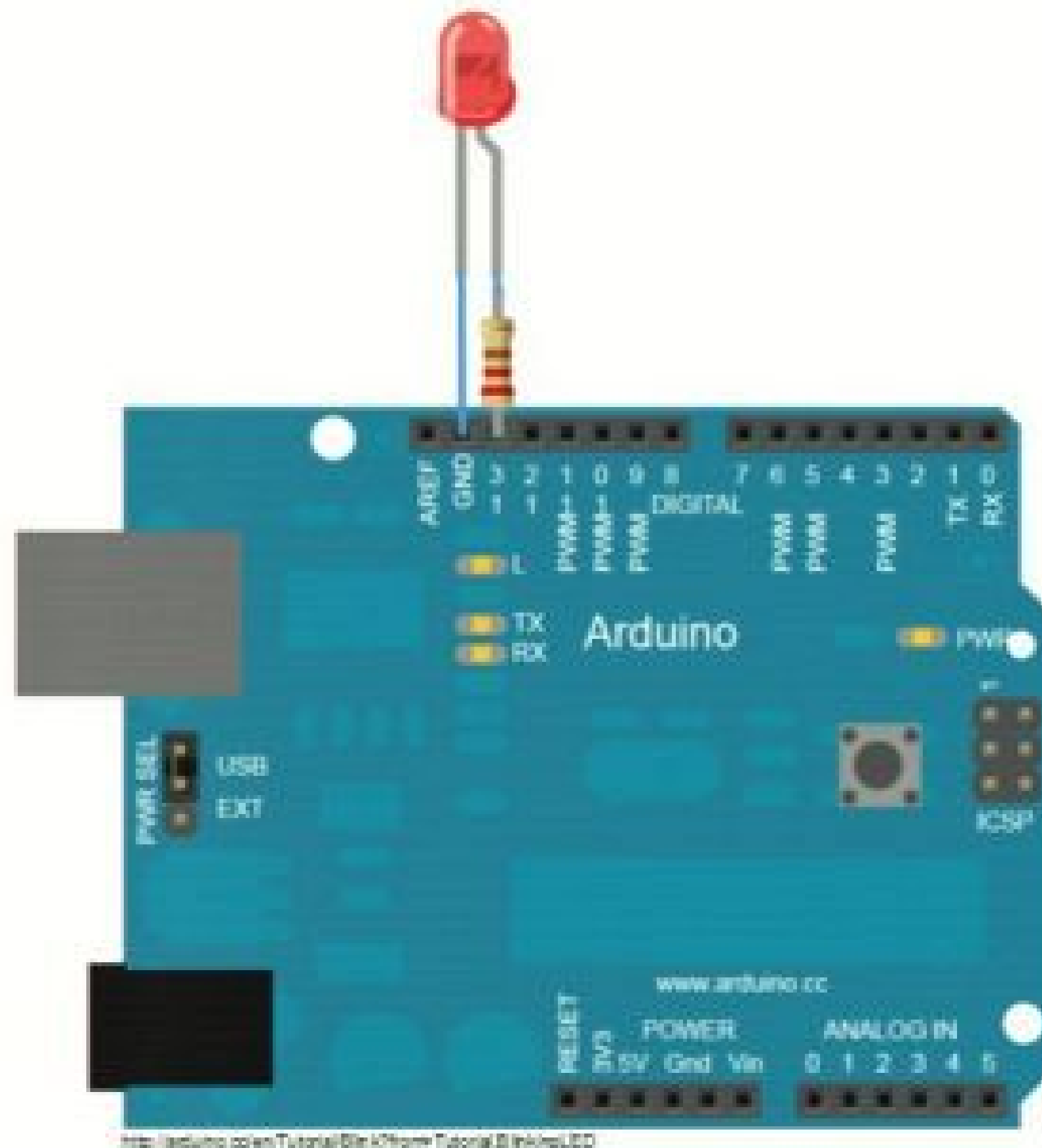


<http://arduino.cc/en/Tutorial/BlinkingLED>

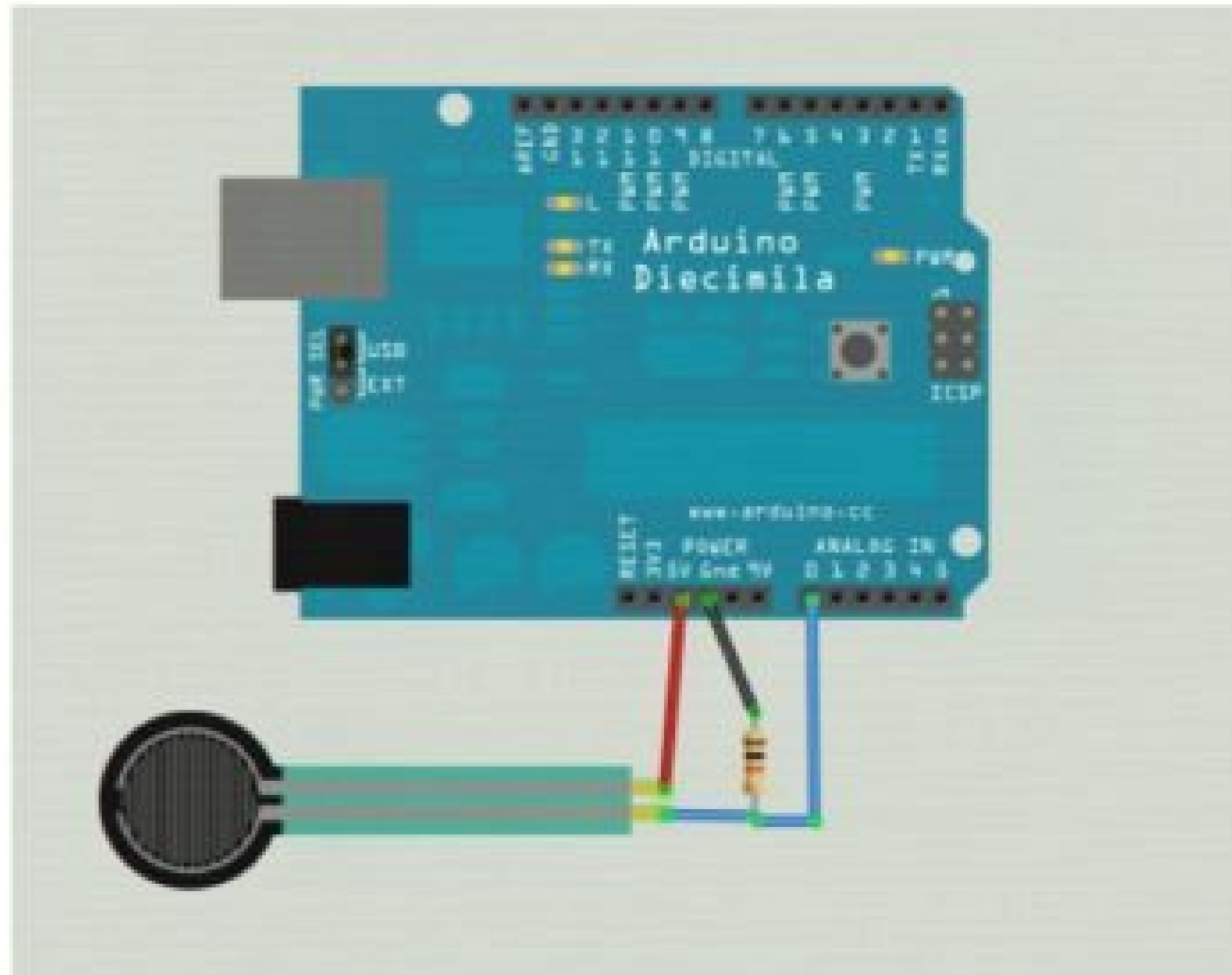


<http://arduino.cc/en/Tutorial/blink> Die Arduino-Plattform: Tutorial: Blinking LED

„Hallo Welt“ für Arduino



Dateneingang und -verarbeitung



<http://www.radioshack.com/500-ohm-potentiometer-for-arduino-301>

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe



```
sketch_jun14a:  
int LEDPIN = 11;  
  
void setup() {  
  pinMode(LEDPIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LEDPIN, HIGH);  
  delay(1000);  
  digitalWrite(LEDPIN, LOW);  
  delay(1000);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

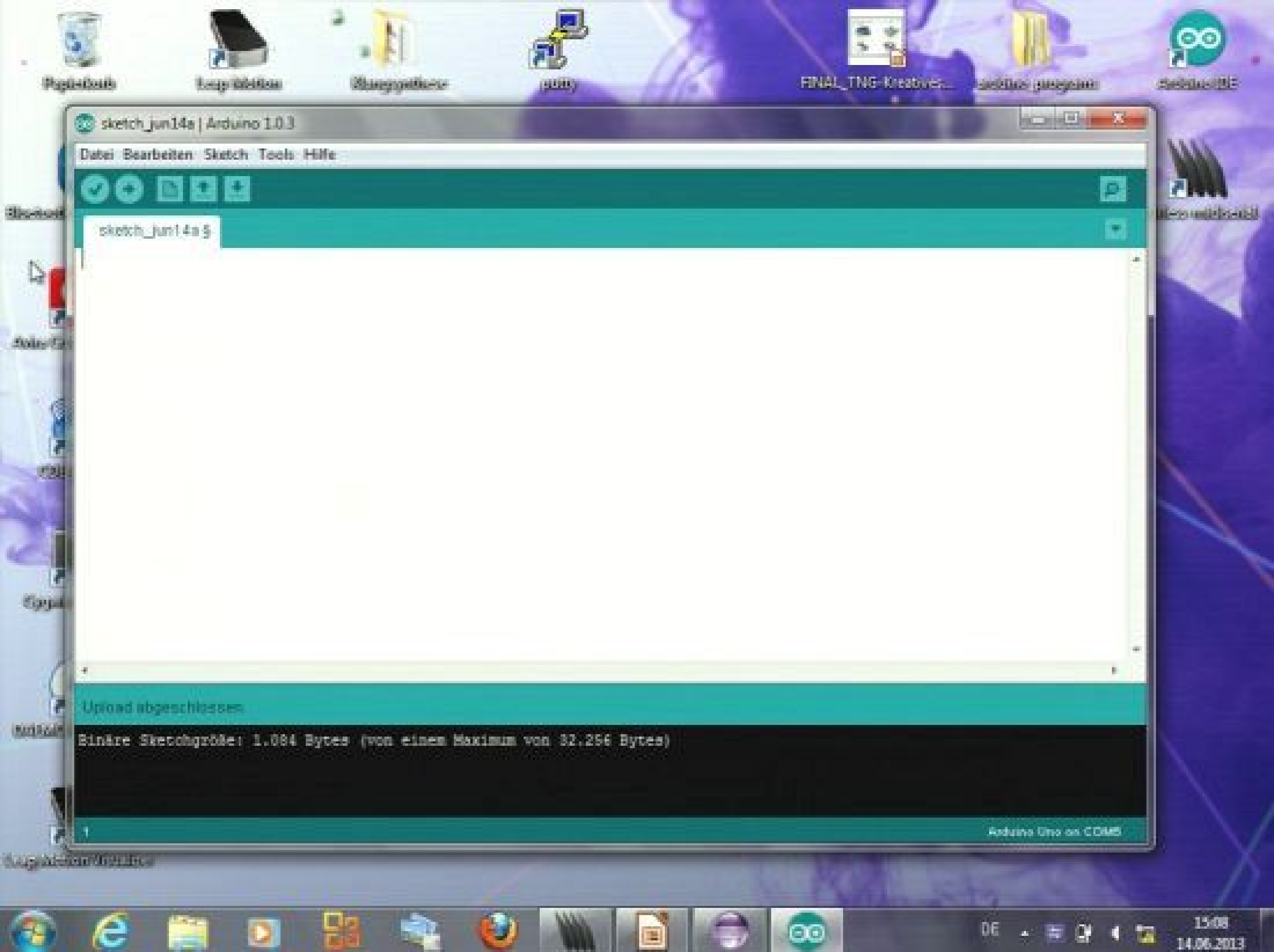
```
sketch_jun14a $  
int LEDPIN = 11;  
  
void setup() {  
  pinMode(LEDPIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LEDPIN, HIGH);  
  delay(1000);  
  digitalWrite(LEDPIN, LOW);  
  delay(1000);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

12

Arduino Uno on COM5



sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe



sketch_jun14a \$

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

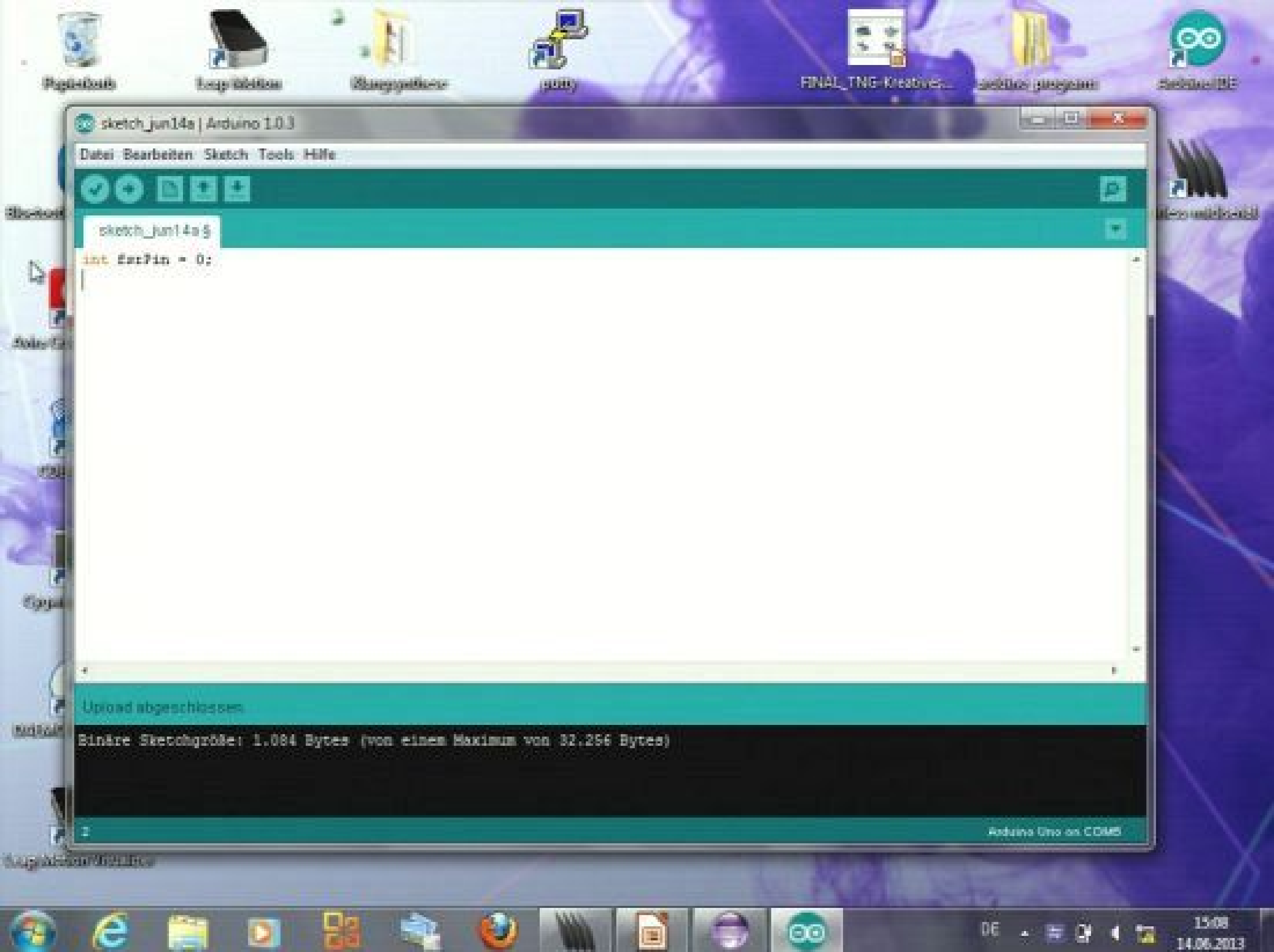
Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Dabei Bearbeiten Sketch Tools Hilfe

```
sketch_jun14a $  
int LedPin = 0;  
|
```

Upload abgeschlossen
Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)
Arduino Uno on COM5



sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe



sketch_jun14a \$

```
int ledPin = 0;  
int ledReading;
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

```
sketch_jun14a:
int ledPin = 0;
int ledReading;

void setup() {
```

Upload abgeschlossen
Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

4 Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

```
sketch_jun14a $  
int ledPin = 0;  
int ledReading;  
  
void setup() {  
  Serial.begin
```

Upload abgeschlossen
Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)
5
Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

```
sketch_jun14a $  
int ledPin = 0;  
int ledReading;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop{
```

Upload abgeschlossen
Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)
Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

```
sketch_jun14a$  
int ledPin = 0;  
int ledReading;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  ledReading;  
}
```

Upload abgeschlossen
Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)
Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Dabei Bearbeiten Sketch Tools Hilfe

```
sketch_jun14a $  
int ledPin = 0;  
int ledReading;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  ledReading = analogRead(0);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

sketch_jun14a \$

```
int EarPin = 0;  
int EarReading;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  EarReading = analogRead(EarPin);  
  Serial.  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

10

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Dabei Bearbeiten Sketch Tools Hilfe



```
sketch_jun14a $  
int EarPin = 0;  
int EarReading;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  EarReading = analogRead(EarPin);  
  Serial.print(EarReading);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

10

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

sketch_jun14a\$

```
int EarPin = 0;  
int EarReading;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  EarReading = analogRead(EarPin);  
  Serial.print("FSR Reading")  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

10

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

```
sketch_jun14a $
int EarPin = 0;
int EarReading;

void setup() {
  Serial.begin(9600);
}

void loop() {
  EarReading = analogRead(EarPin);
  Serial.print("FSR Reading: ");
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

10 Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

```
sketch_jun14a $
int EncPin = 0;
int EncReading;

void setup() {
  Serial.begin(9600);
}

void loop() {
  EncReading = analogRead(EncPin);
  Serial.print("FSR Reading: ");
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Dabei Bearbeiten Sketch Tools Hilfe

sketch_jun14a\$

```
int fscPin = 0;
```

```
int fscReading;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
  fscReading = analogRead(fscPin);
```

```
  Serial.print("FSC Reading: ");
```

```
  Serial.println;
```

```
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

11

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Dabei Bearbeiten Sketch Tools Hilfe

sketch_jun14a \$

```
int fcrPin = 0;  
int fcrReading;  
  
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  fcrReading = analogRead(fcrPin);  
  Serial.print("FSR Reading: ");  
  Serial.println(fcrReading);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 1.084 Bytes (von einem Maximum von 32.256 Bytes)

12

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe

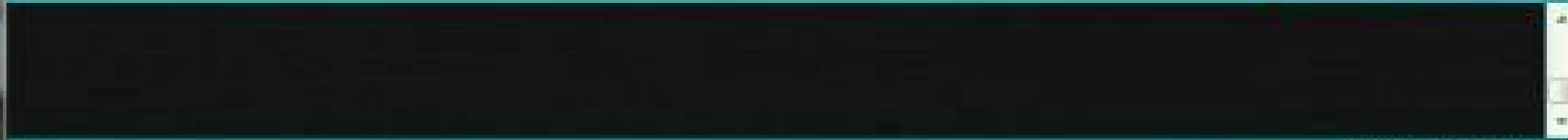


```
sketch_jun14a:
int fscPin = 0;
int fscReading;

void setup() {
  Serial.begin(9600);
}

void loop() {
  fscReading = analogRead(fscPin);
  Serial.print("FSC Reading: ");
  Serial.println(fscReading);
  delay(500);
}
```

Sketch kompilieren



12 Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe



sketch_jun14a \$

```
int fcrPin = 0;
int fcrReading;

void setup() {
  Serial.begin(9600);
}

void loop() {
  fcrReading = analogRead(fcrPin);
  Serial.print("FSR Reading: ");
  Serial.println(fcrReading);
  delay(500);
}
```

Kompilierung abgeschlossen

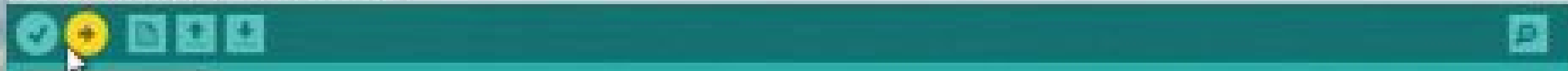
Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12

Arduino Uno on COM5

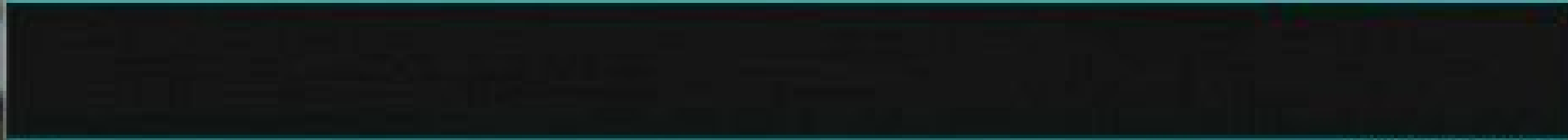
sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe



```
sketch_jun14a:  
int ledPin = 0;  
int ledReading;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  ledReading = analogRead(ledPin);  
  Serial.print("LED Reading: ");  
  Serial.println(ledReading);  
  delay(500);  
}
```

Sketch kompilieren...



12 Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe



sketch_jun14a \$

```
int fcrPin = 0;
int fcrReading;

void setup() {
  Serial.begin(9600);
}

void loop() {
  fcrReading = analogRead(fcrPin);
  Serial.print("FSR Reading: ");
  Serial.println(fcrReading);
  delay(500);
}
```

Uploaden



Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12

Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe



sketch_jun14a \$

```
int fcrPin = 0;
int fcrReading;

void setup() {
  Serial.begin(9600);
}

void loop() {
  fcrReading = analogRead(fcrPin);
  Serial.print("FSR Reading: ");
  Serial.println(fcrReading);
  delay(500);
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12

Arduino Uno on COM5

Senden

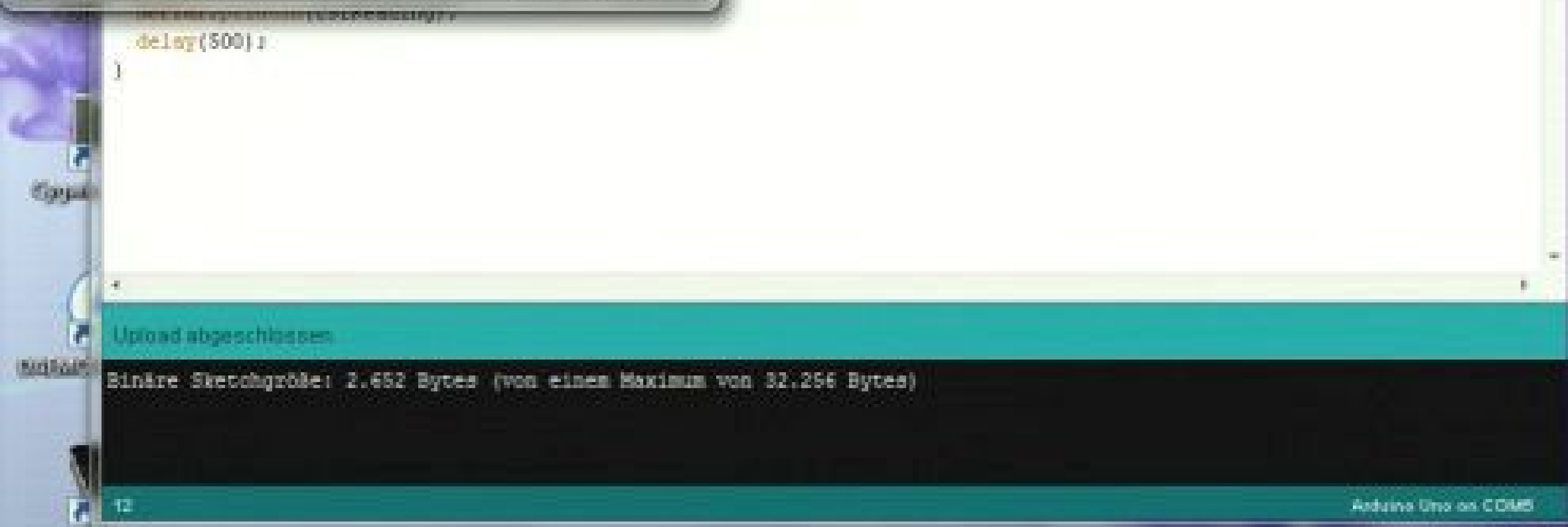
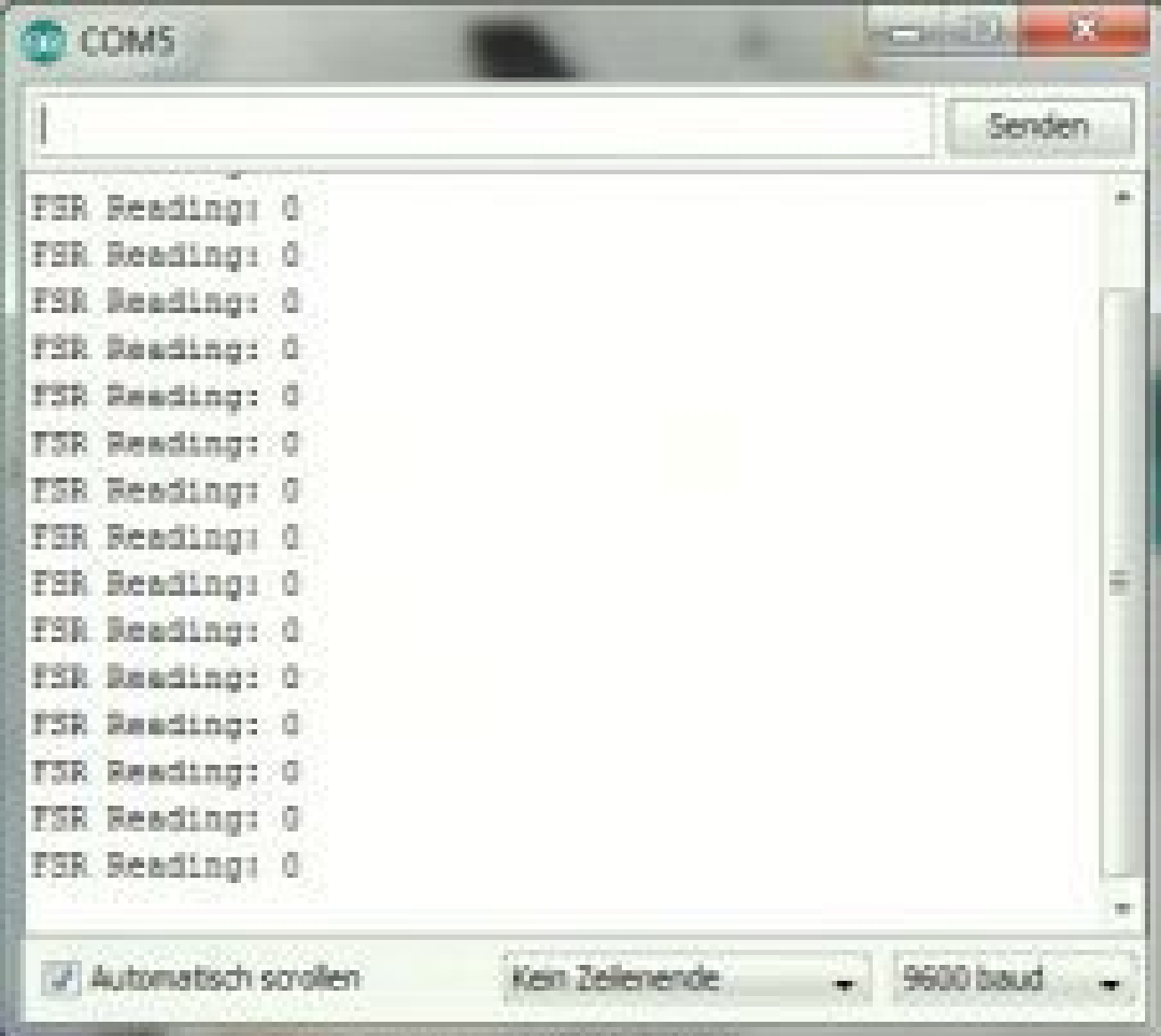
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0



☒ Automatisch scrolen

Kein Zeilenende

9600 baud



COM5

Senden

FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 35
FSR Reading: 98
FSR Reading: 208
FSR Reading: 379

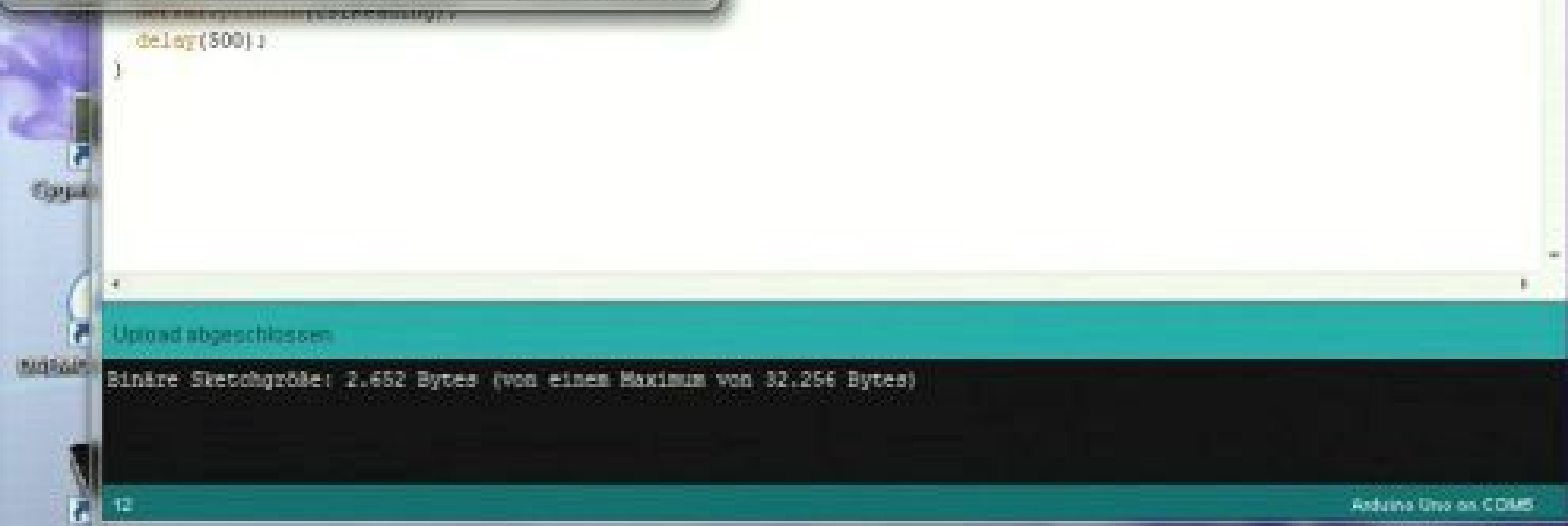
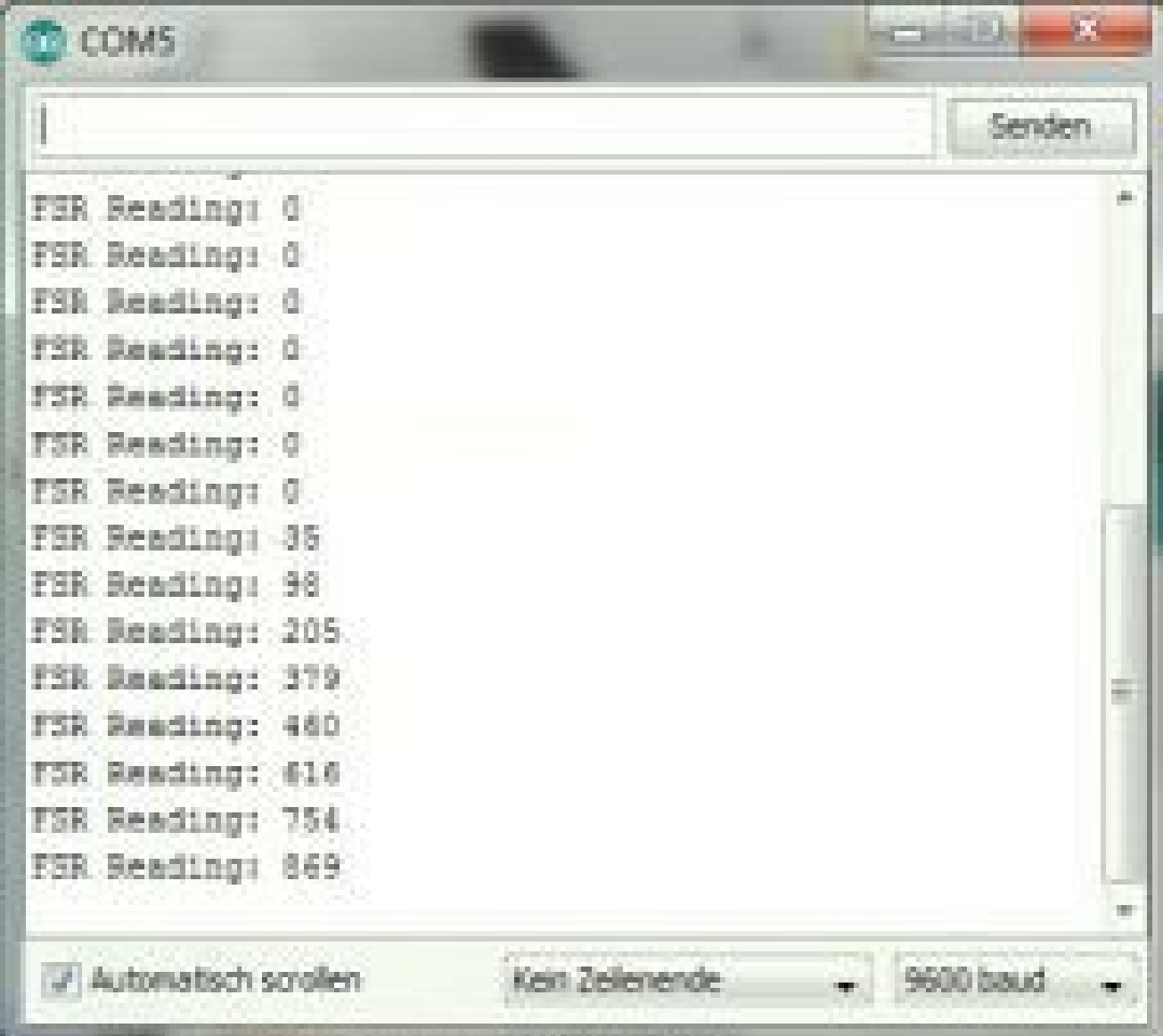
☒ Automatisch scrollen Kein Zeilenende 9600 baud

```
delay(500);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12 Arduino Uno on COM5



COM5

Senden

FSR Reading: 379
FSR Reading: 460
FSR Reading: 616
FSR Reading: 754
FSR Reading: 889
FSR Reading: 923
FSR Reading: 976
FSR Reading: 988
FSR Reading: 990
FSR Reading: 993
FSR Reading: 994
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0

☒ Automatisch scrollen Kein Zeilenende 9600 baud

```
delay(500);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12 Arduino Uno on COM5

COM5

Senden

FSR Reading: 869
FSR Reading: 923
FSR Reading: 976
FSR Reading: 988
FSR Reading: 990
FSR Reading: 993
FSR Reading: 994
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 3
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0

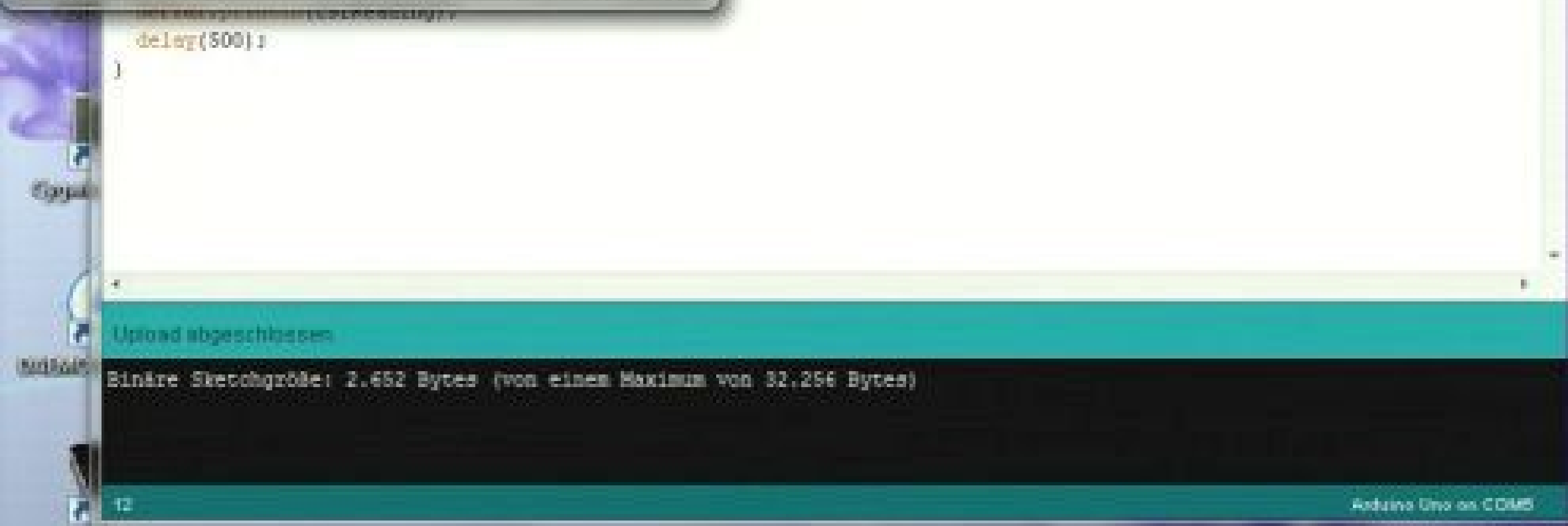
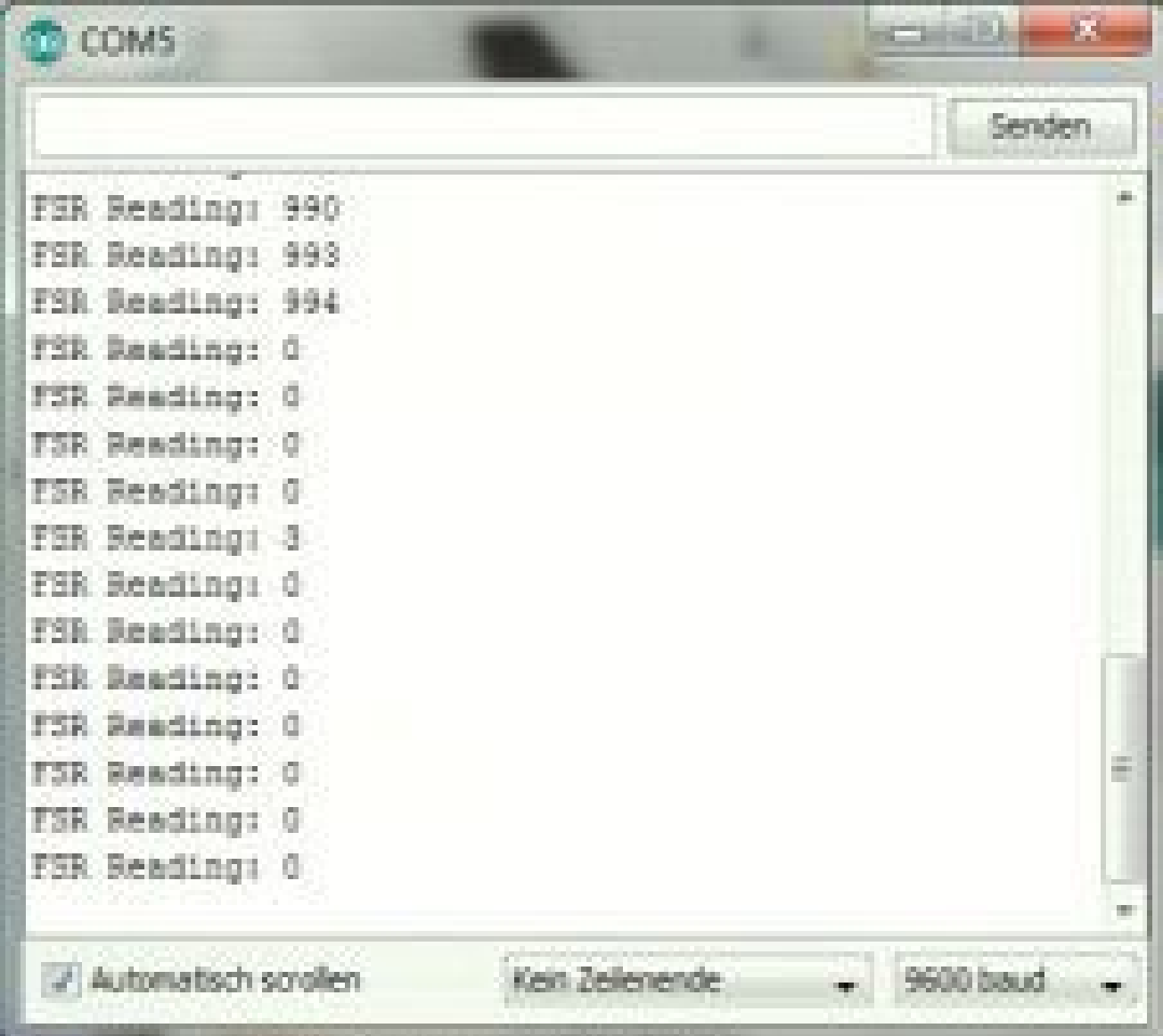
☒ Automatisch scrollen Kein Zeilenende 9600 baud

```
delay(500);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12 Arduino Uno on COM5



COM5

Senden

FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 3
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 356
FSR Reading: 497
FSR Reading: 368

☒ Automatisch scrollen Kein Zeilenende 9600 baud

```
delay(500);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12 Arduino Uno on COM5

COM5

Senden

FSR Reading: 3
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 356
FSR Reading: 497
FSR Reading: 368
FSR Reading: 412
FSR Reading: 808
FSR Reading: 0

☒ Automatisch scrollen Kein Zeilenende 9600 baud

```
void setup() {  
  pinMode(12, OUTPUT);  
  delay(500);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12 Arduino Uno on COM5

COM5

Senden

FSR Reading: 0
FSR Reading: 356
FSR Reading: 497
FSR Reading: 368
FSR Reading: 412
FSR Reading: 888
FSR Reading: 0
FSR Reading: 216
FSR Reading: 0
FSR Reading: 166
FSR Reading: 5
FSR Reading: 132
FSR Reading: 280
FSR Reading: 0
FSR Reading: 0

☒ Automatisch scrollen ☐ Kein Zeilenende 9600 baud

```
void setup() {  
  pinMode(2, OUTPUT);  
  delay(500);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12 Arduino Uno on COM5

COM5

Senden

FSR Reading: 166
FSR Reading: 5
FSR Reading: 132
FSR Reading: 280
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0

☒ Automatisch scrolen Kein Zeilenende 9600 baud

```
void setup() {  
  pinMode(10, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(10, HIGH);  
  delay(500);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12 Arduino Uno on COM5

COM5

Senden

FSR Reading: 280
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Reading: 0
FSR Read

☒ Automatisch scrollen ☐ Kein Zeilenende 9600 baud

```
delay(500);  
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12 Arduino Uno on COM5

sketch_jun14a | Arduino 1.0.3

Dabei Bearbeiten Sketch Tools Hilfe

sketch_jun14a \$

```
int farPin = 0;
```

```
int farReading;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
  farReading = analogRead(farPin);
```

```
  Serial.print("FSR Reading: ");
```

```
  Serial.println(farReading);
```

```
  delay(500);
```

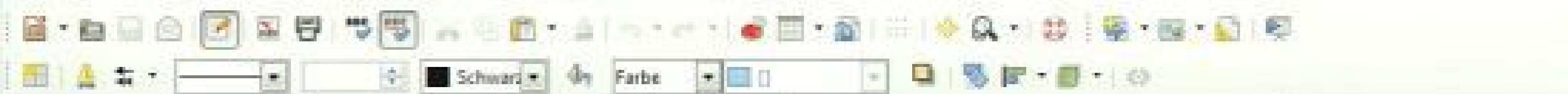
```
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.652 Bytes (von einem Maximum von 32.256 Bytes)

12

Arduino Uno on COM5



Folien

Normal Gliederung Notizen Handzettel Foliensortierung

Aufgabenbereich Anzeige



Einführung in die Arduino-Plattform

TNG TECHNOLOGY CONSULTING

Dateneingang und -verarbeitung

20130614_Kreatives_Hardware_Hacking_mit_der_Arduino_Plattform 18

Masterfolien

Layouts

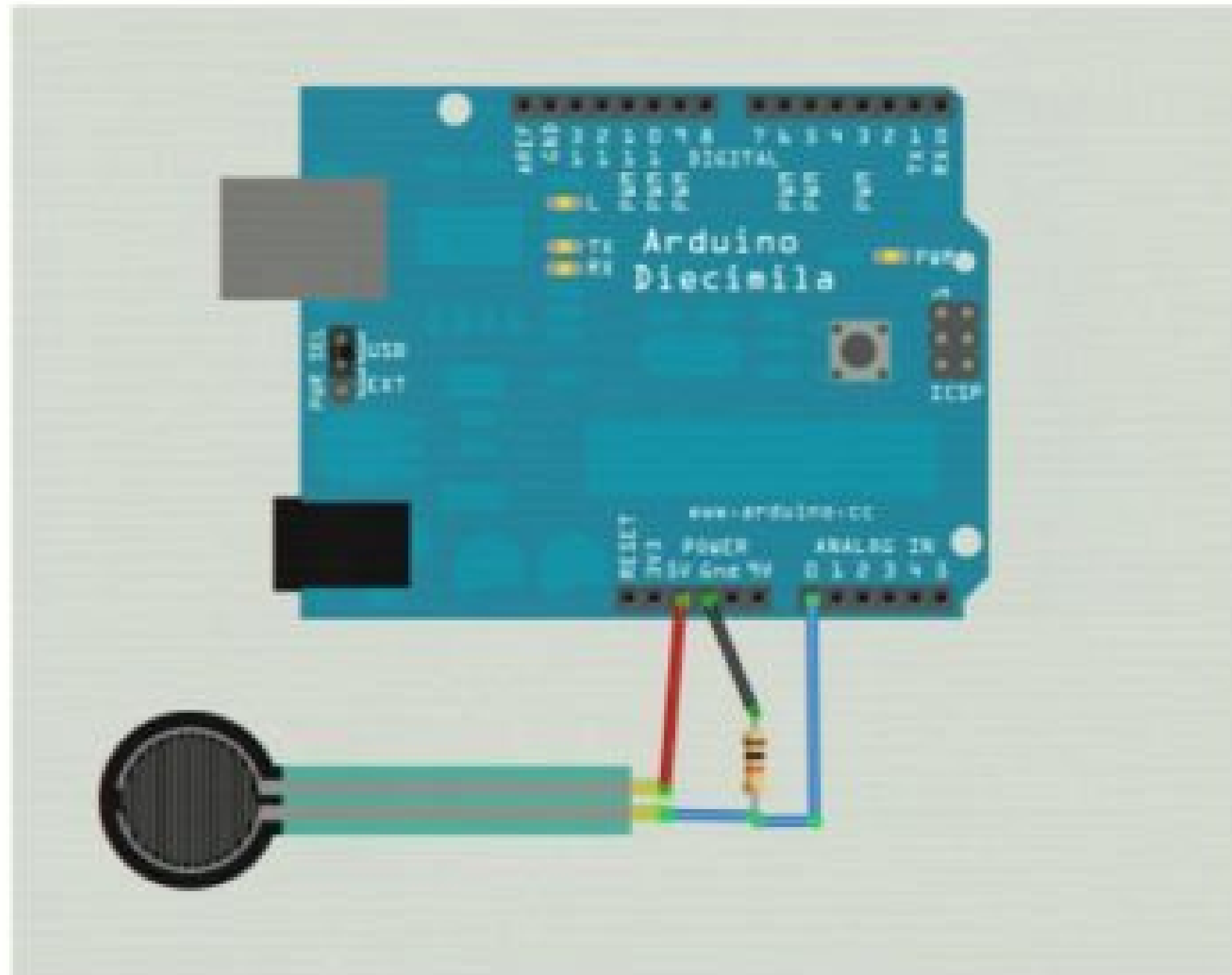
Tabellen

Benutzerdefinierte Animation

Folienubergang



Dateneingang und -verarbeitung



<http://www.radiopk.com/5k-ohm-potentiometer-for-arduino/>

Agenda

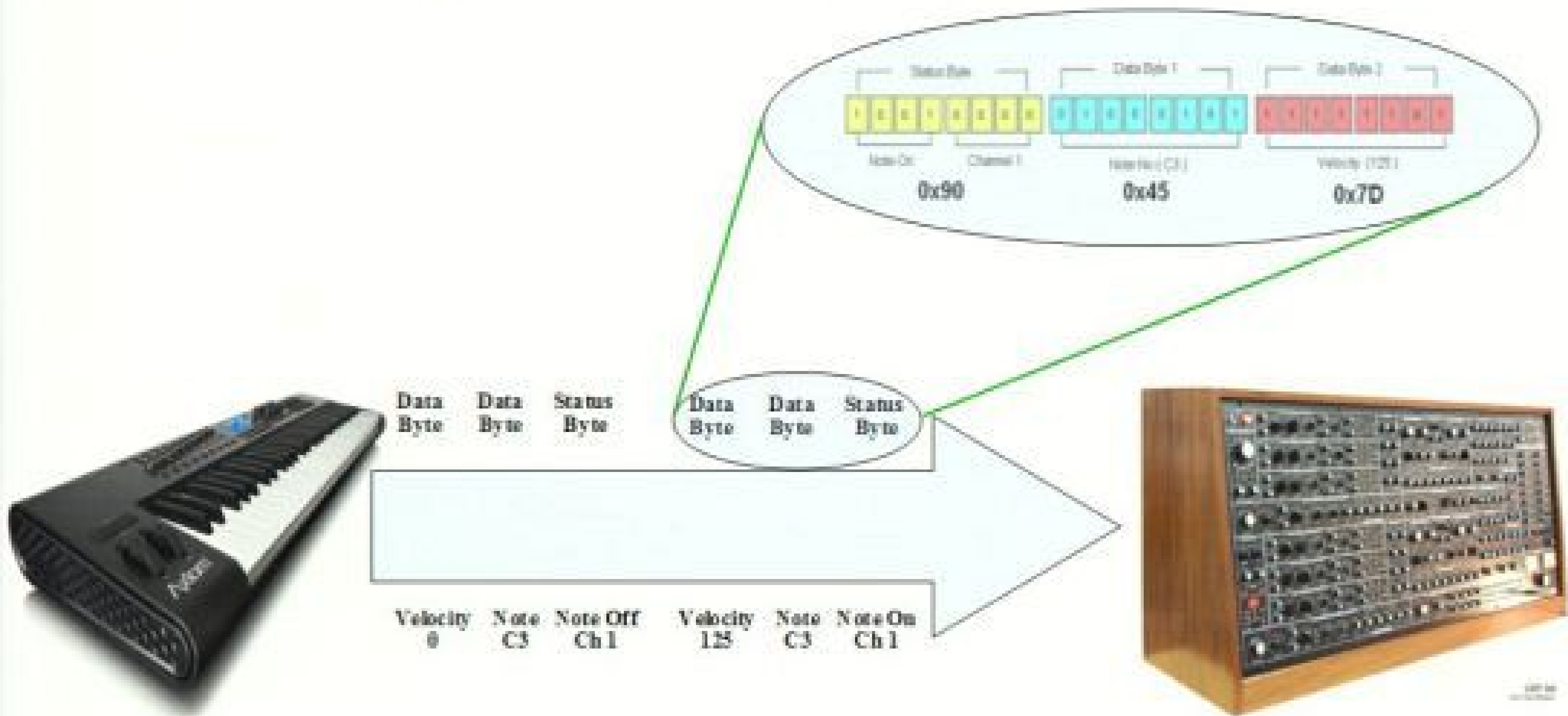
- Einführung in die Arduino-Plattform
- **Was ist das – MIDI?**
- Beat Sneakers
- Laser Theremin
- Lavalampe
- Der menschliche Körper als Controller LEAP

Es war einmal, vor 31 Jahren

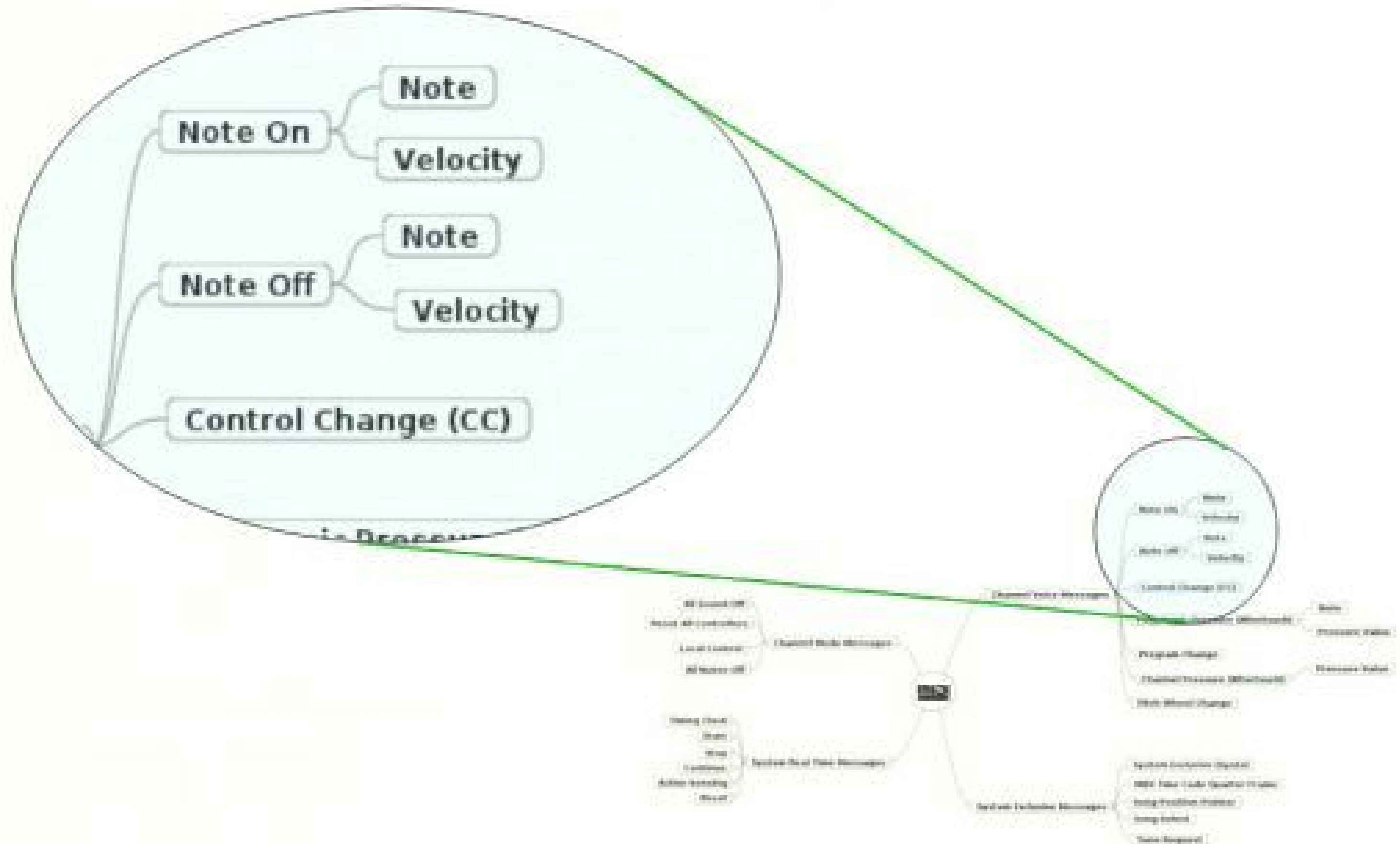


- Digitale Schnittstelle für Musikinstrumente
- 1982: MIDI v1.0
- Erfinder:
 - Dave Smith
 - Ikutaro Kakehashi (Roland Corporation)
- Ursprünglich zu Interkommunikation von Synthesizern entwickelt
- Erweiterungen u.a. durch **General MIDI**

Von Status- und Data-Bytes



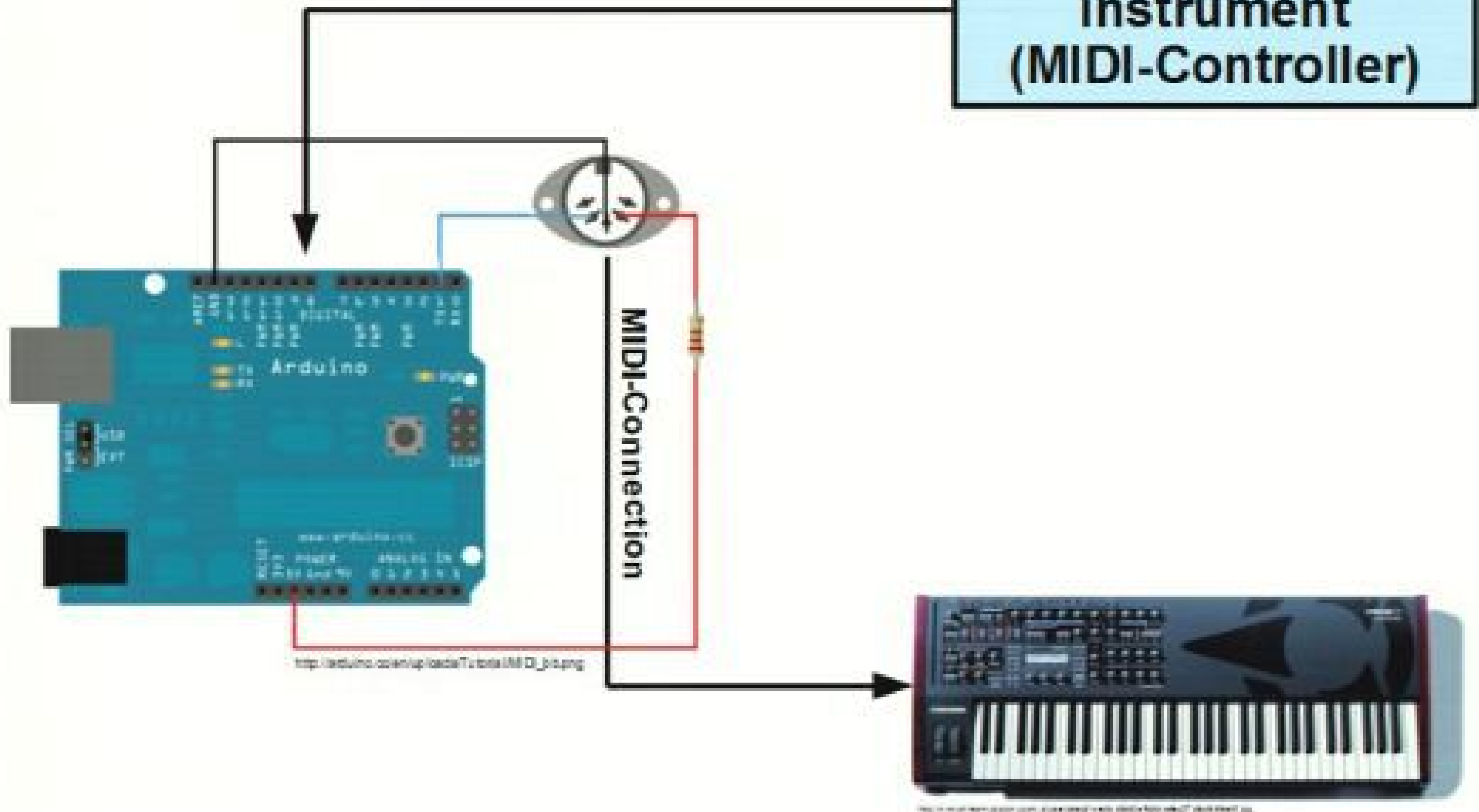
Daten über Daten



Direct MIDI

MIDI

**Instrument
(MIDI-Controller)**



MIDI-Serial Bridge



**Instrument
(MIDI-Controller)**

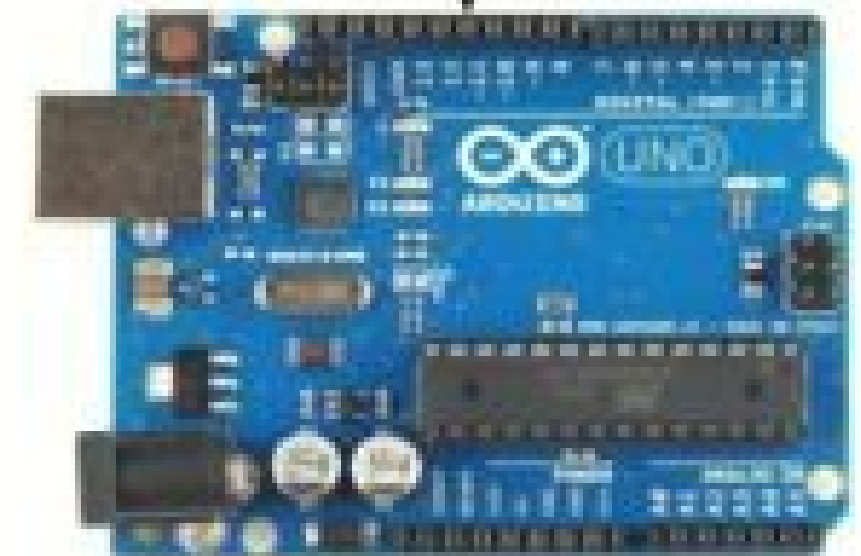


<http://content.sll.de.com/900/1005/596/132.jpg>

serial connection



MIDI-Signale



Arduino Uno

USB-MIDI connection



MIDI-Signale



Synthesizer

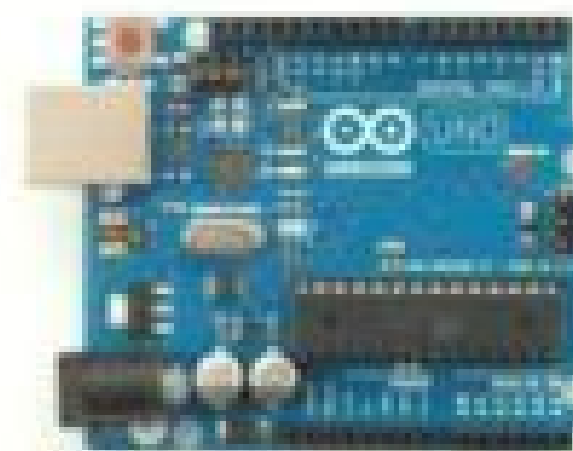
Agenda

- Einführung in die Arduino-Plattform
- Was ist das – MIDI?
- **Beat Sneakers**
- Laser Theremin
- Lavalampe
- Der menschliche Körper als Controller LEAP

Zutatenliste



<http://www.sneakers.com/beat-sneakers.html>



<http://www.arduino.cc/en/Main/ArduinoBoardUno>



<http://www.keychain.com/keychain.com>



<http://www.coinbase.com/quarter>

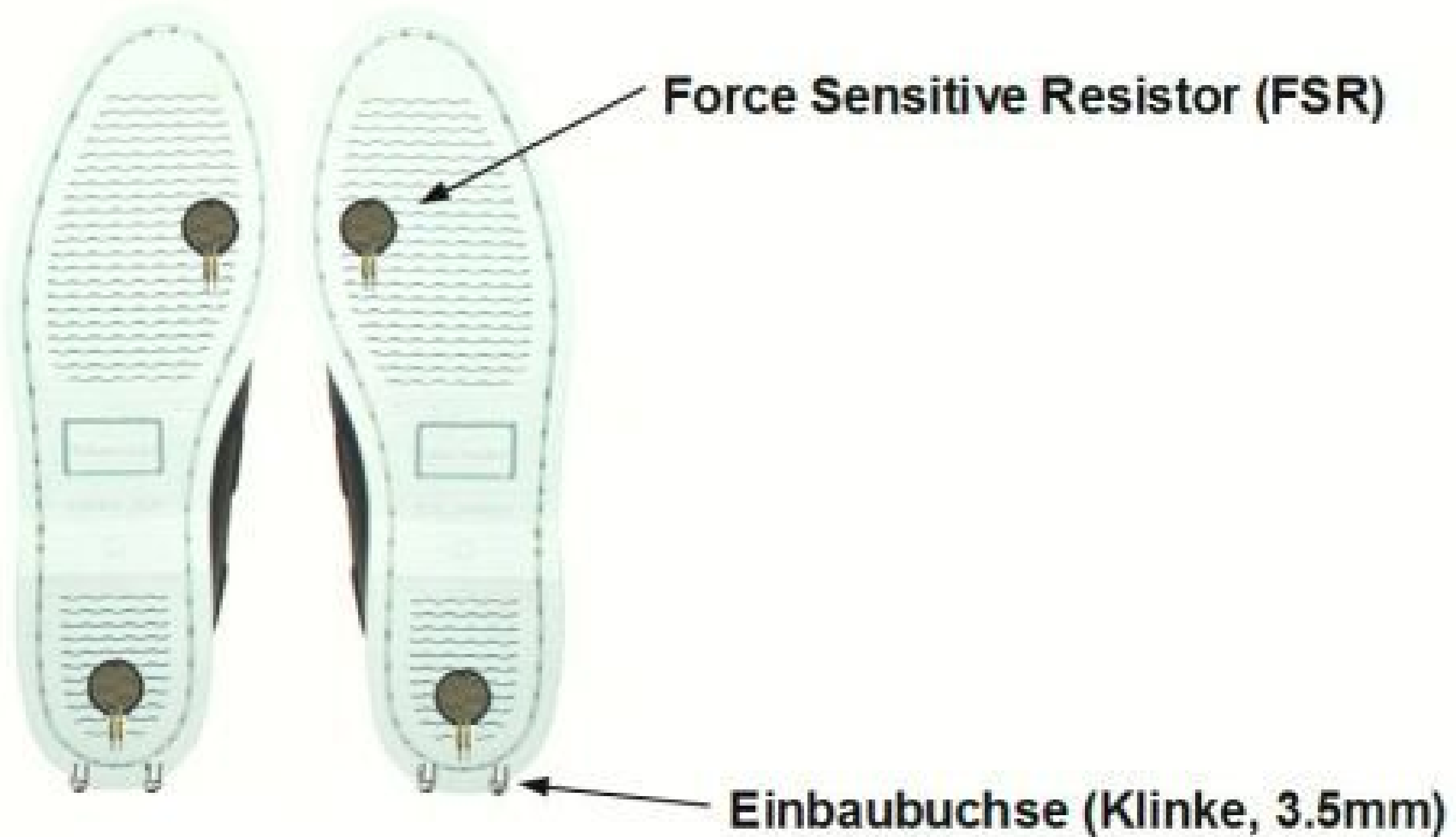


<http://www.jumper.com/jumper.com>

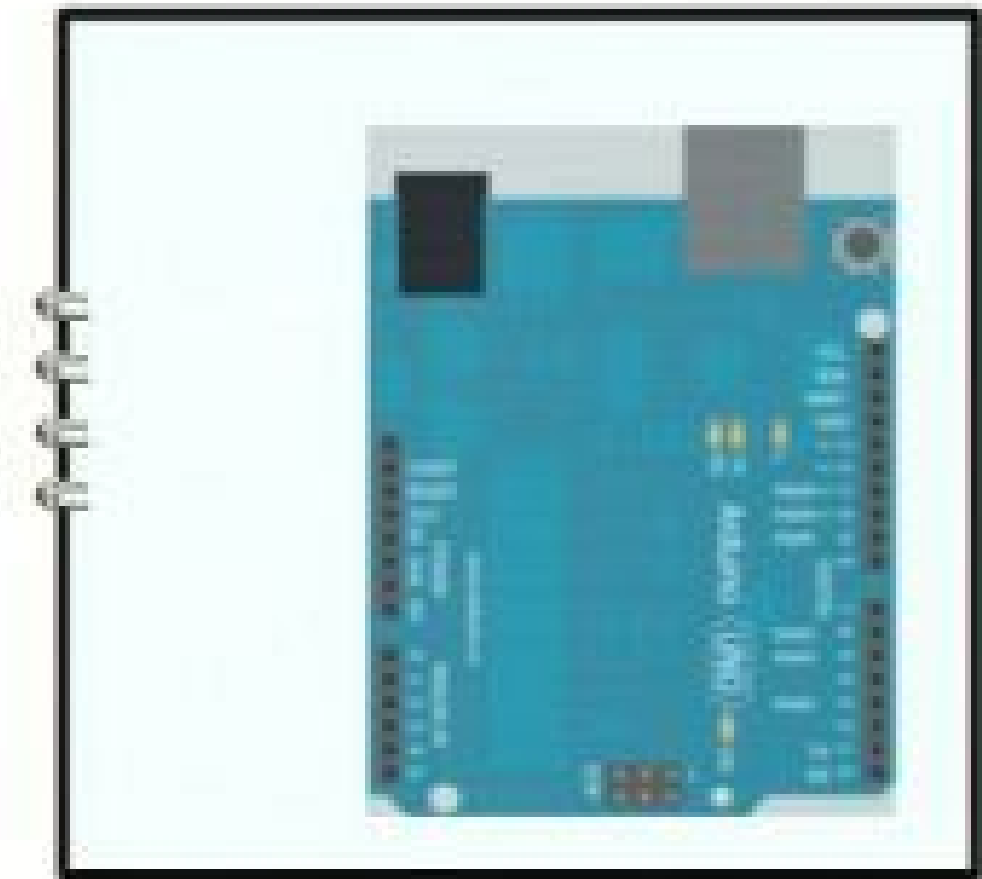
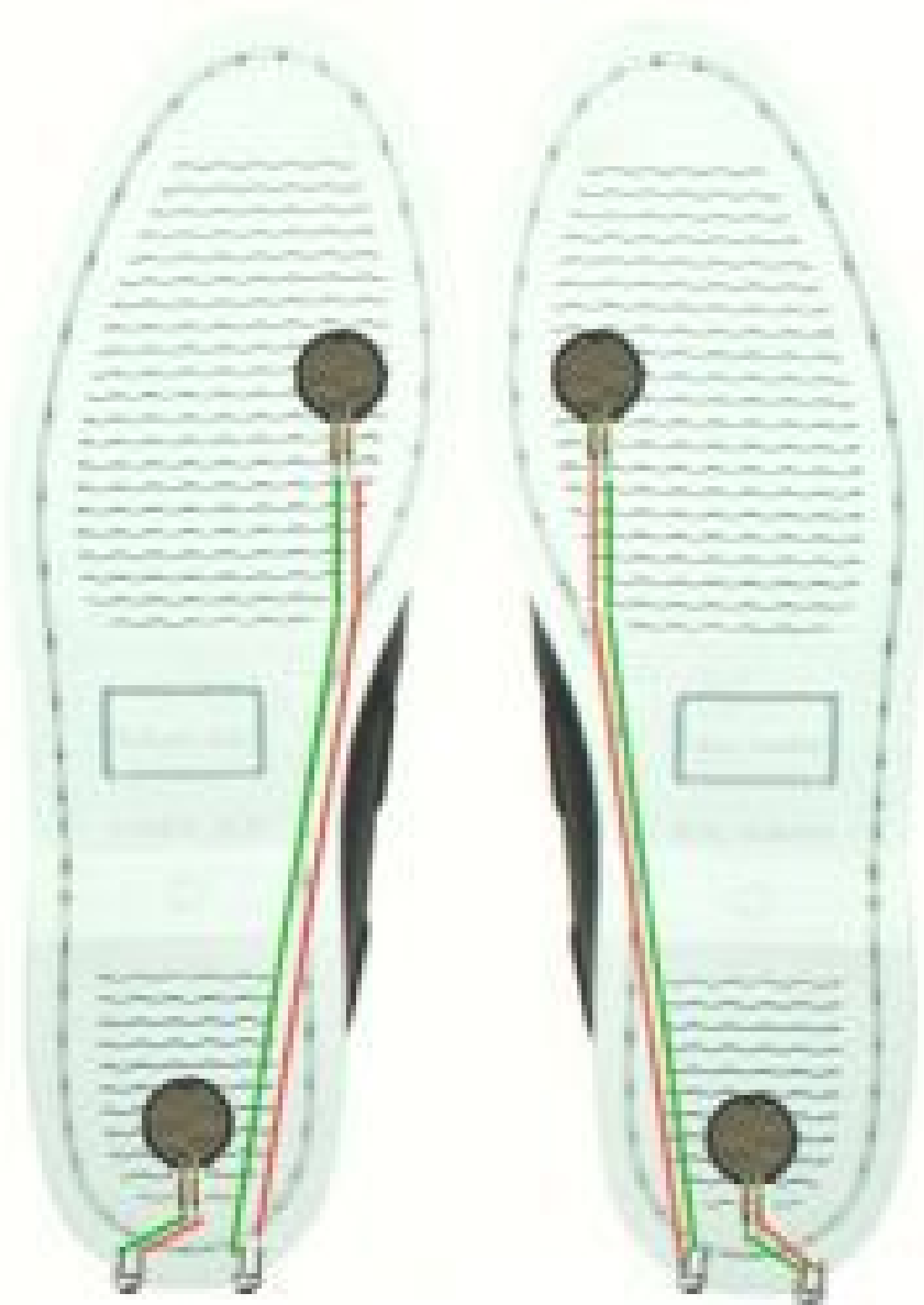


<http://www.3m.com/3m/us/en>

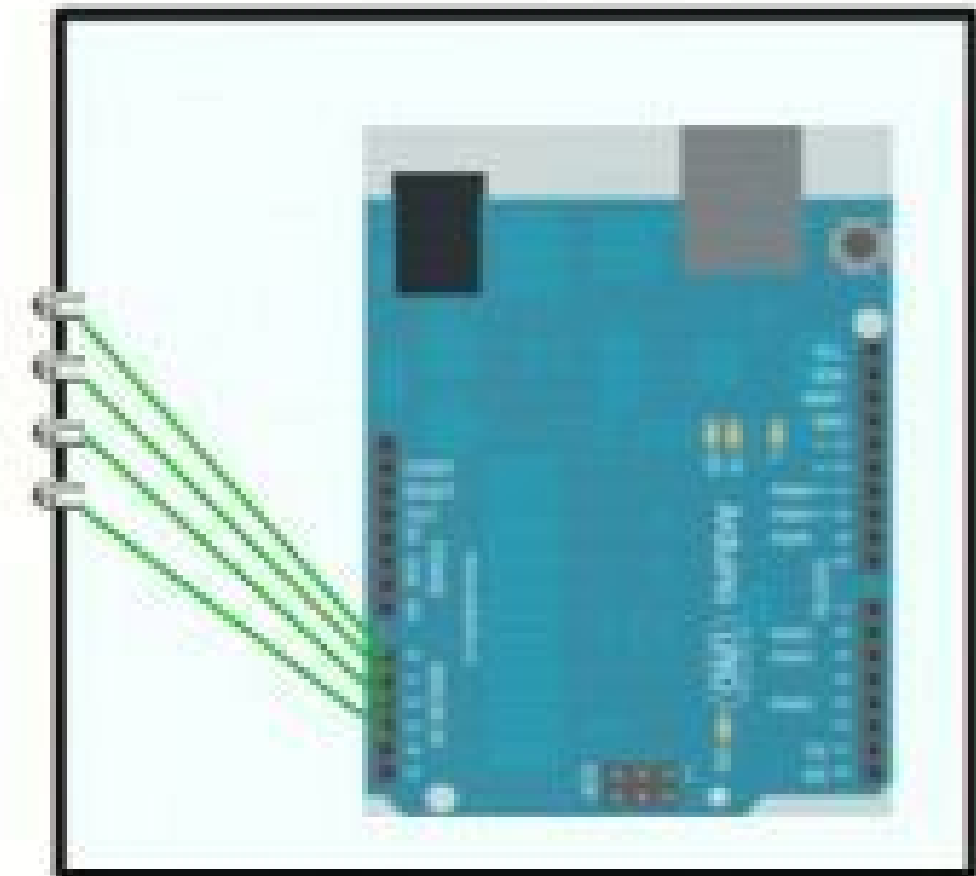
Zuerst die Sensoren



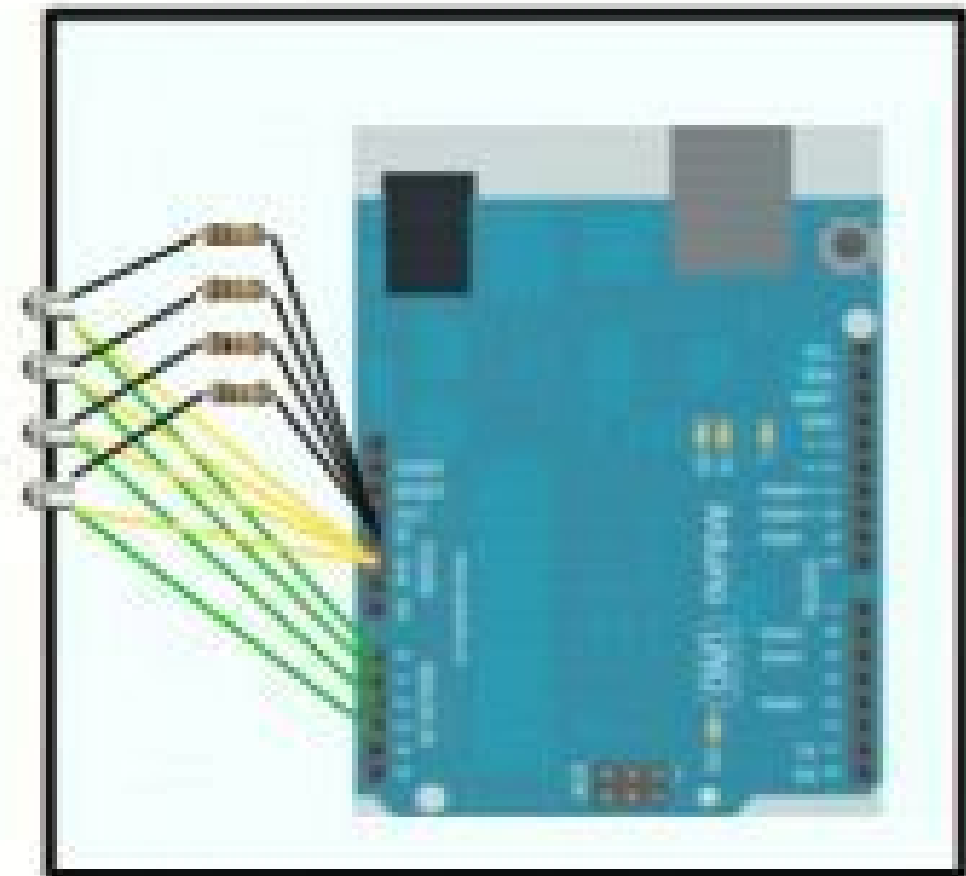
Arduino bereit legen



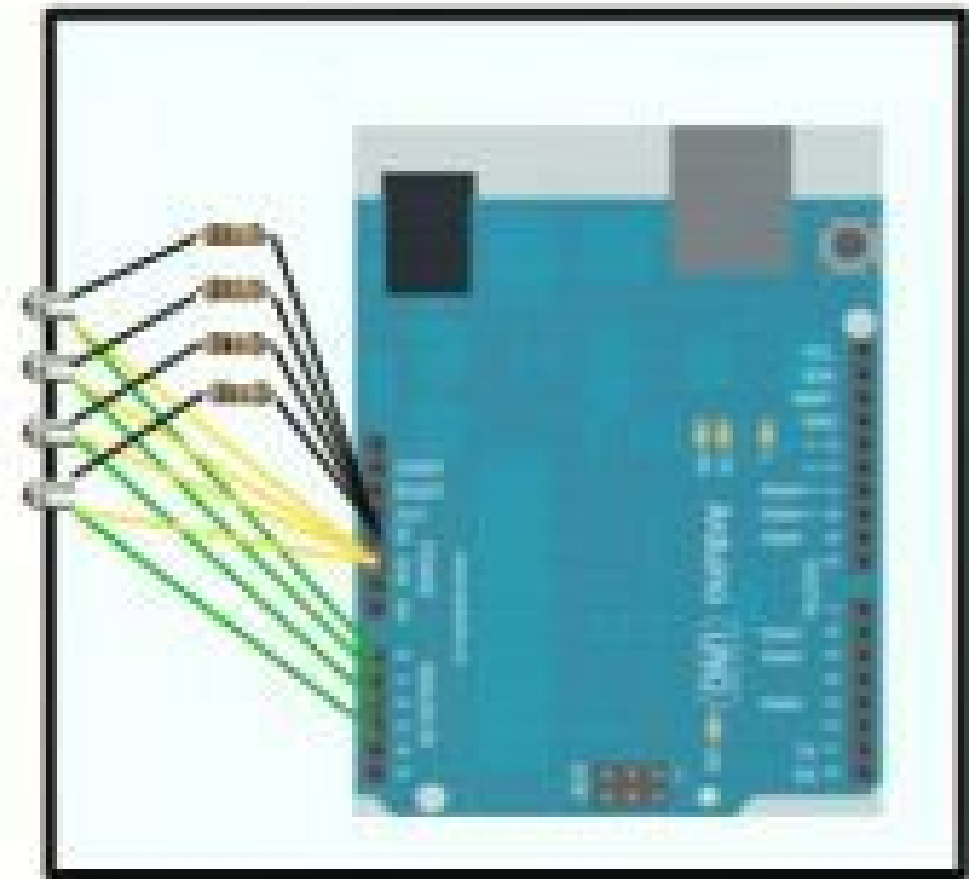
Die Datenleitung für Analogports



Spannung kommt an die andere Seite



Nun noch Beides verbinden



Die Schuhsoftware

```

#include <Arduino.h>

#define HEALTHTHRESHOLD 250
#define TOOTHRESHOLD 250
#define LOWERHEALTHTHRESHOLD 230
#define LOWERTOOTHRESHOLD 100
#define HEELPIN 0
#define HEELPIN 1
#define TOEPIN 2
#define TOEPIN 3
#define CHANNEL 1

int LEDpin = 5;
int pinNote[4] = {1, 2, 3, 4};
boolean heel1 = false;
boolean heel2 = false;
boolean toe1 = false;
boolean toe2 = false;

void setup() {
  MIDI.begin(1);
  pinMode(LEDpin, OUTPUT);
}

void loop() {
  heel1 = sender(analogRead(HEELPIN), HEELPIN, pinNote[HEELPIN], "HEEL1", heel1, HEALTHTHRESHOLD, LOWERHEALTHTHRESHOLD);
  heel2 = sender(analogRead(HEELPIN), HEELPIN, pinNote[HEELPIN], "HEEL2", heel2, HEALTHTHRESHOLD, LOWERHEALTHTHRESHOLD);
  toe1 = sender(analogRead(TOEPIN)+100, TOEPIN, pinNote[TOEPIN], "TOE1", toe1, TOOTHRESHOLD, LOWERTOOTHRESHOLD);
  toe2 = sender(analogRead(TOEPIN)+100, TOEPIN, pinNote[TOEPIN], "TOE2", toe2, TOOTHRESHOLD, LOWERTOOTHRESHOLD);
}

boolean sender(int reading, int pin, int note, char eng1, boolean press, int threshold, int lowthreshold) {
  if(reading == threshold) {
    digitalWrite(LEDpin, LOW);
    reading = reading/9 - 1;
    digitalWrite(LEDpin, reading);
    MIDI.sendNoteOn(note, reading, CHANNEL);
    delay(300);
    MIDI.sendNoteOff(note, 0, CHANNEL);
    digitalWrite(LEDpin, LOW);
  }
  press = true;
}
else if(reading == lowthreshold) {
  press = false;
}
return press;
}

```

Empfindlichkeit der Sensoren

MIDI-Standardnoten

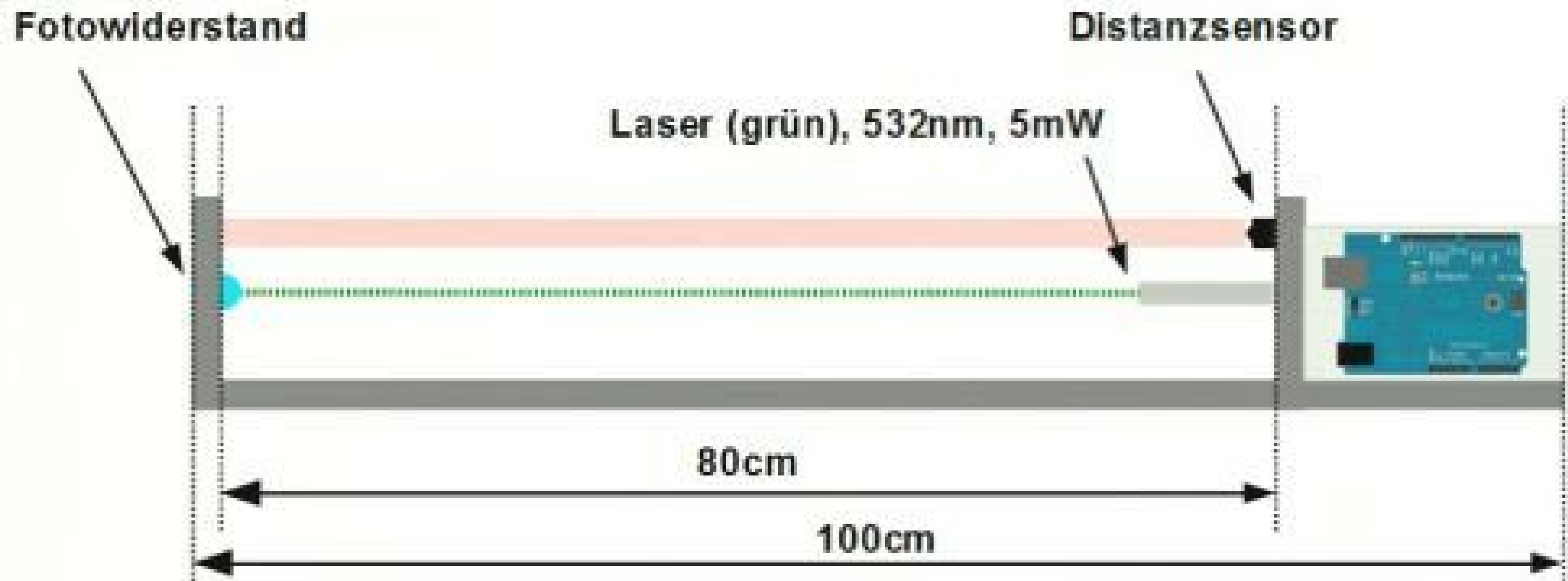
Abfragen der einzelnen Sensoren

MIDI-Nachrichten

Agenda

- Einführung in die Arduino-Plattform
- Was ist das – MIDI?
- Beat Sneakers
- **Laser Theremin**
- Lavalampe
- Der menschliche Körper als Controller LEAP

Aufbau



Aufbau

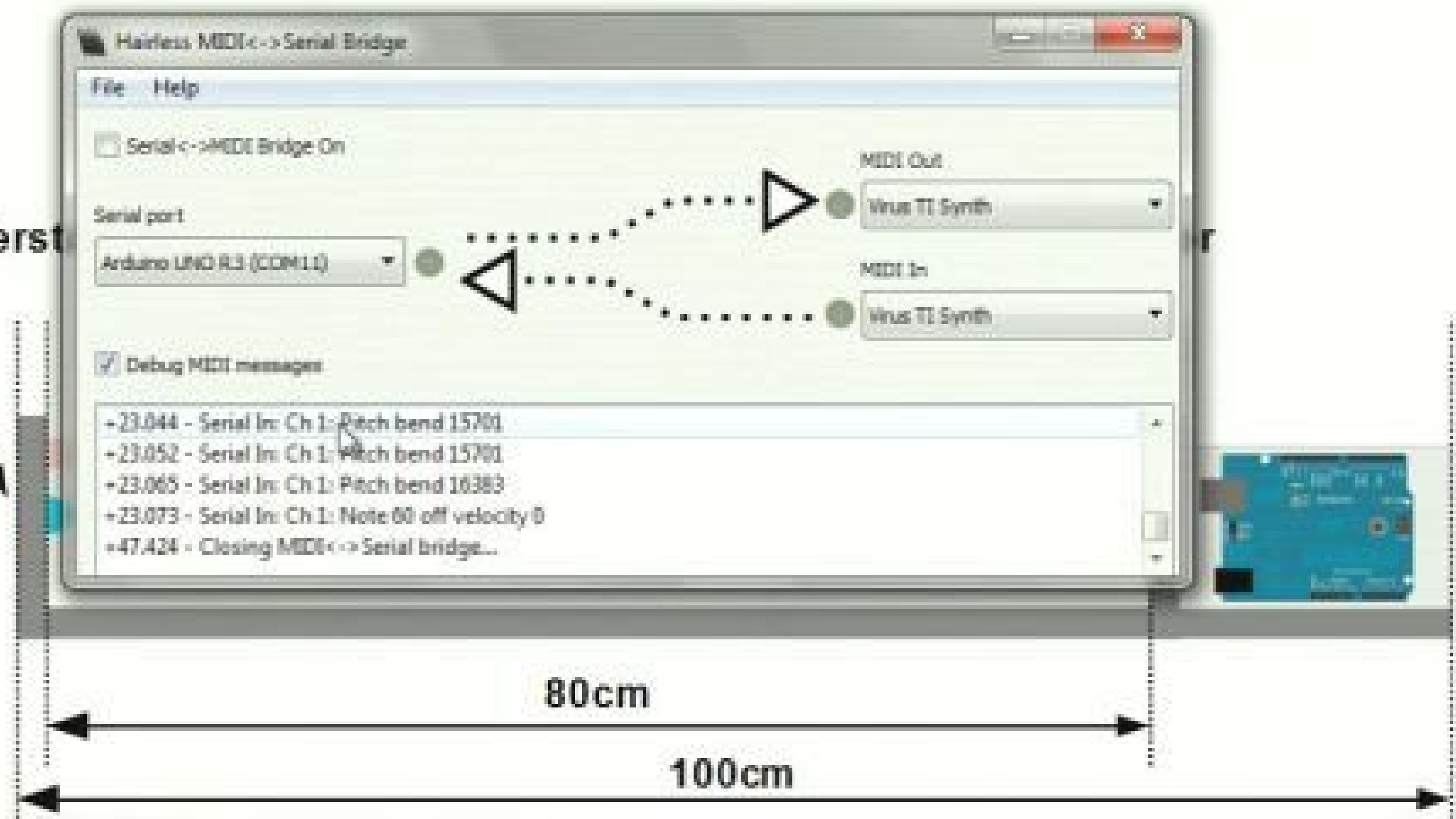
Fotowiderstand

Distanzsensord



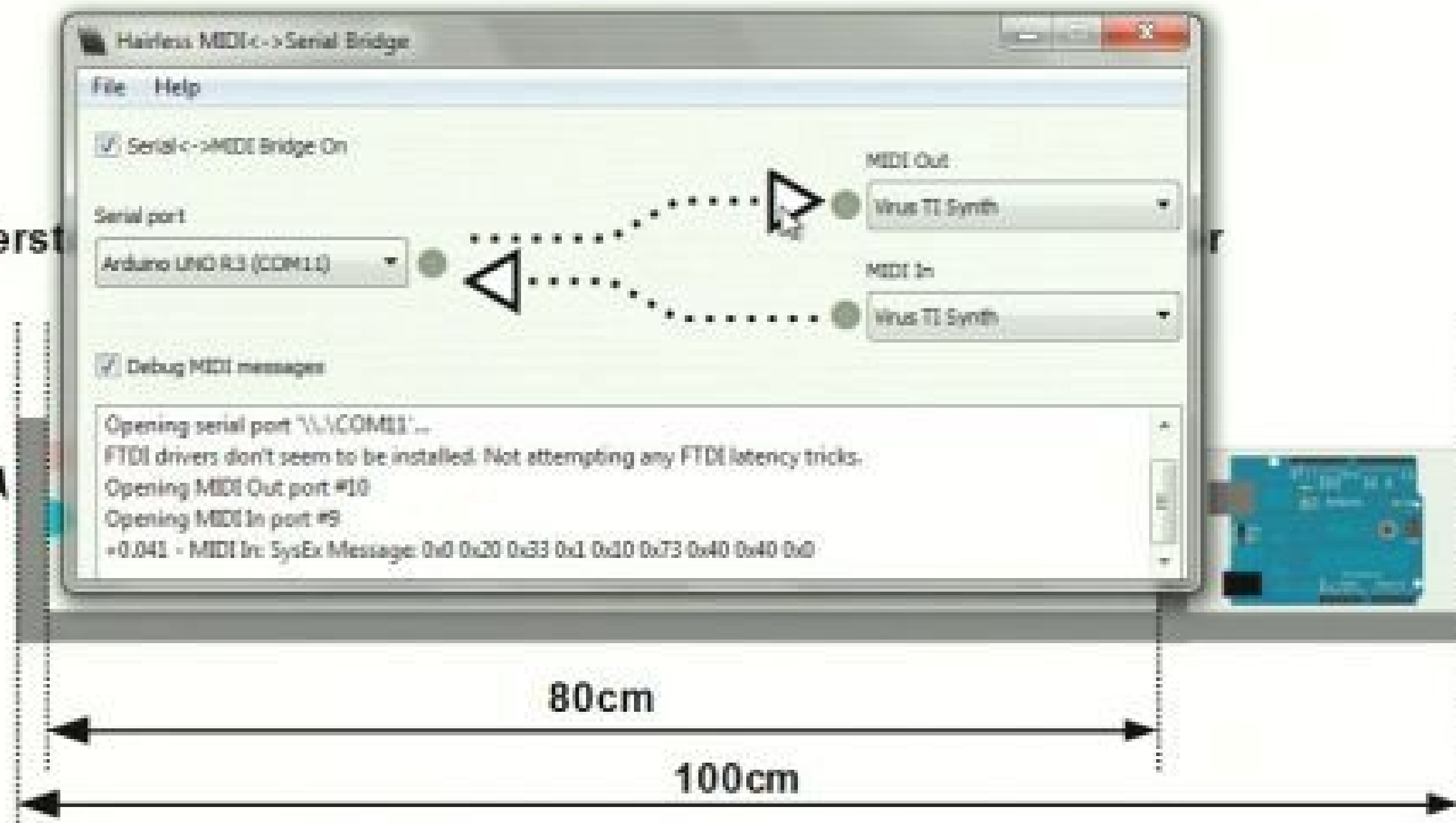
Aufbau

Fotowiderst



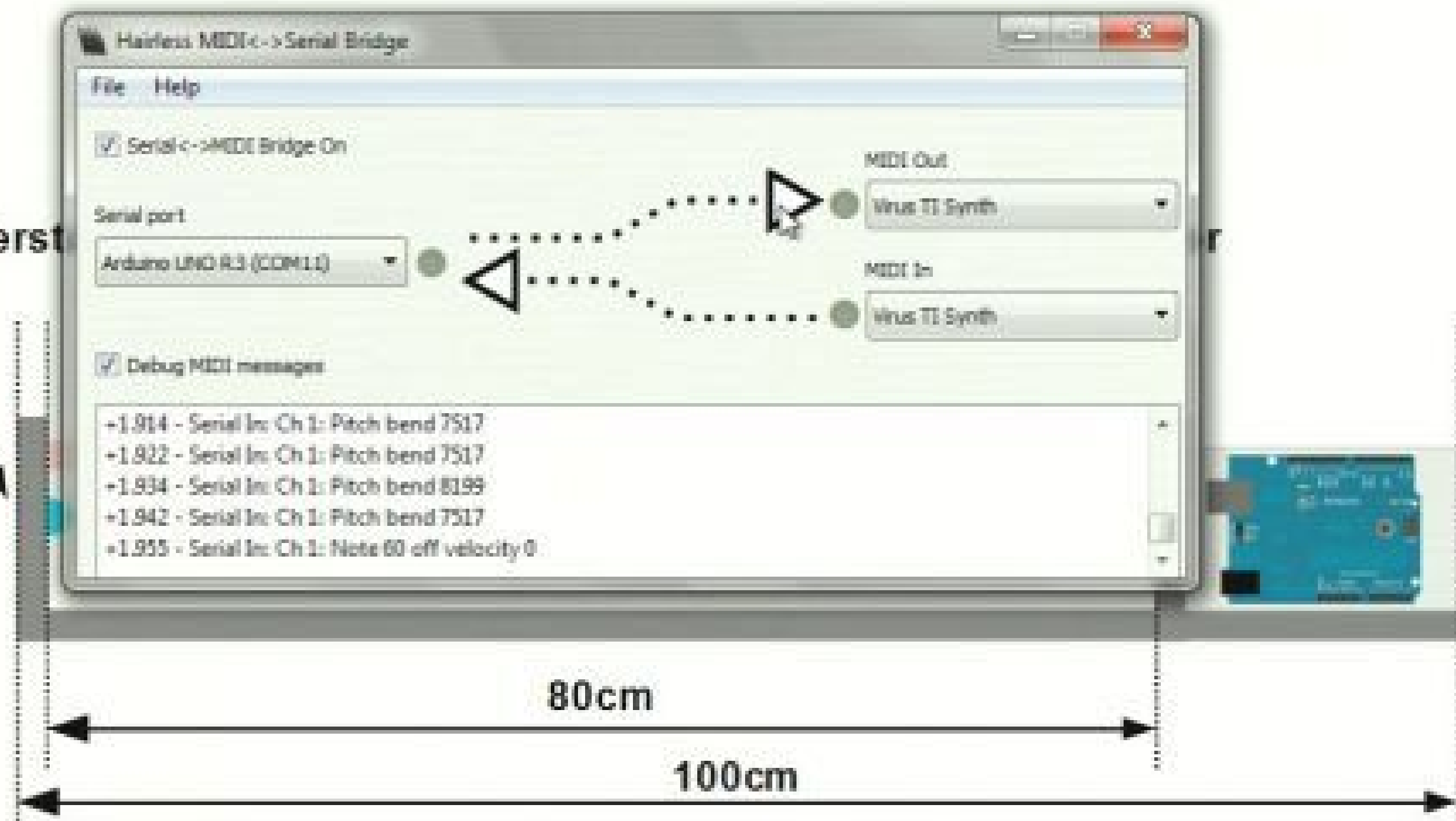
Aufbau

Fotowiderst



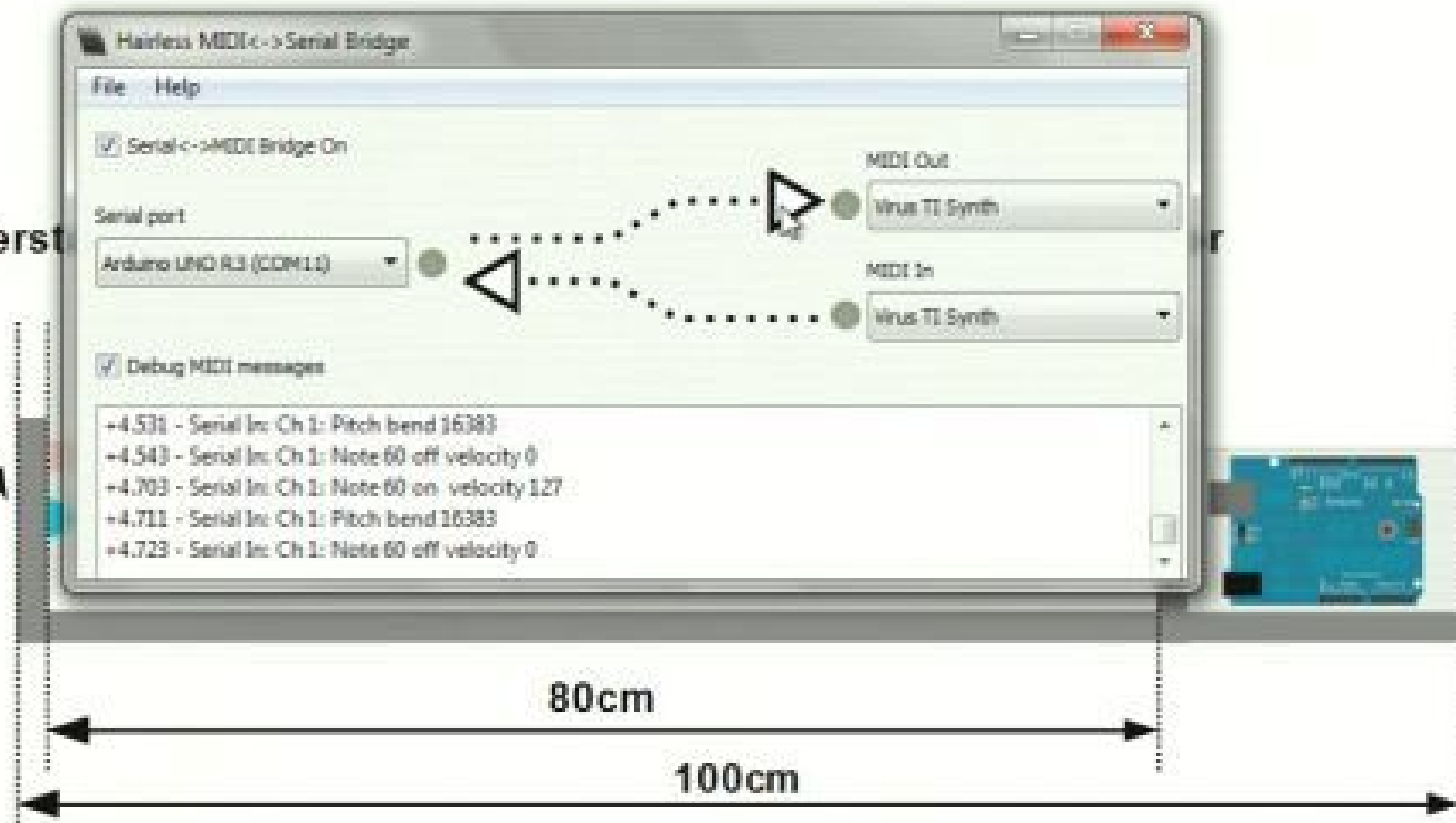
Aufbau

Fotowiderst

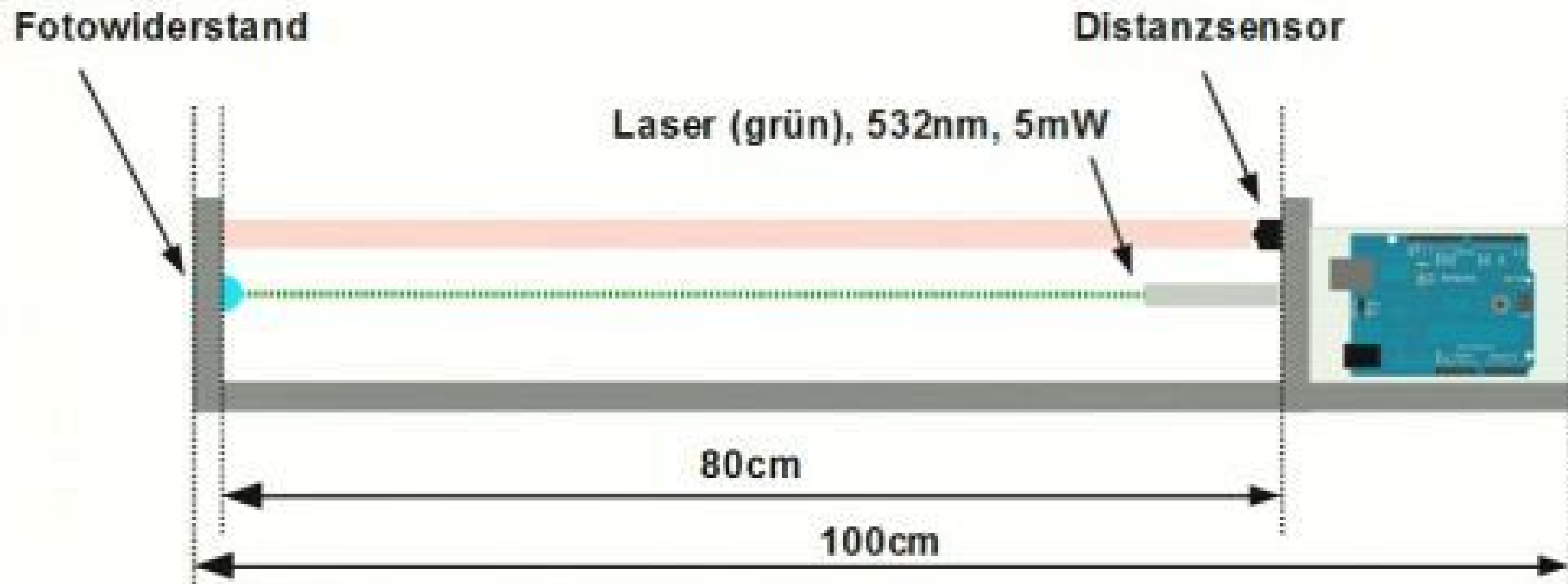


Aufbau

Fotowiderst



Aufbau



Aufbau

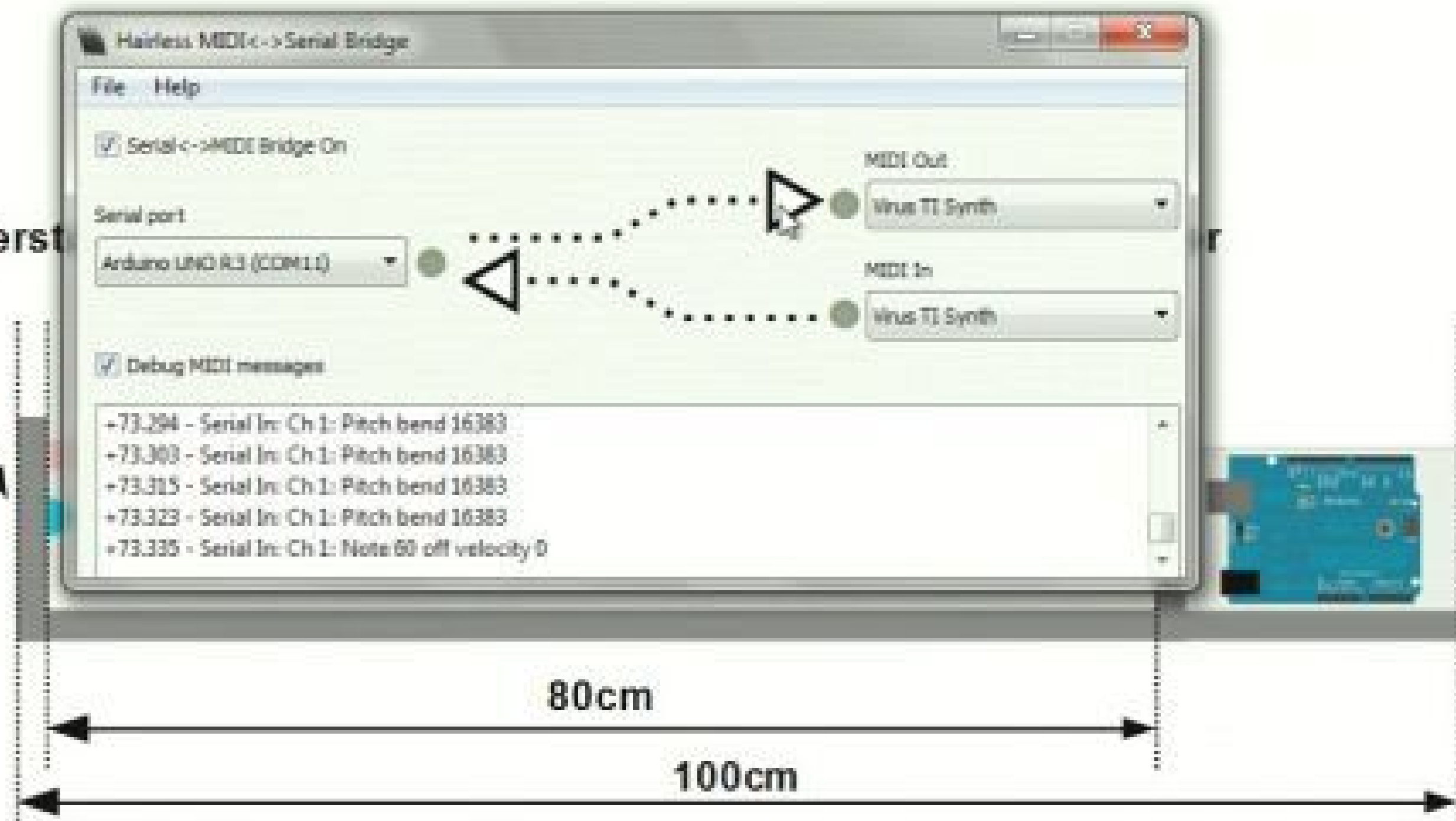
Fotowiderstand

Distanzsensord



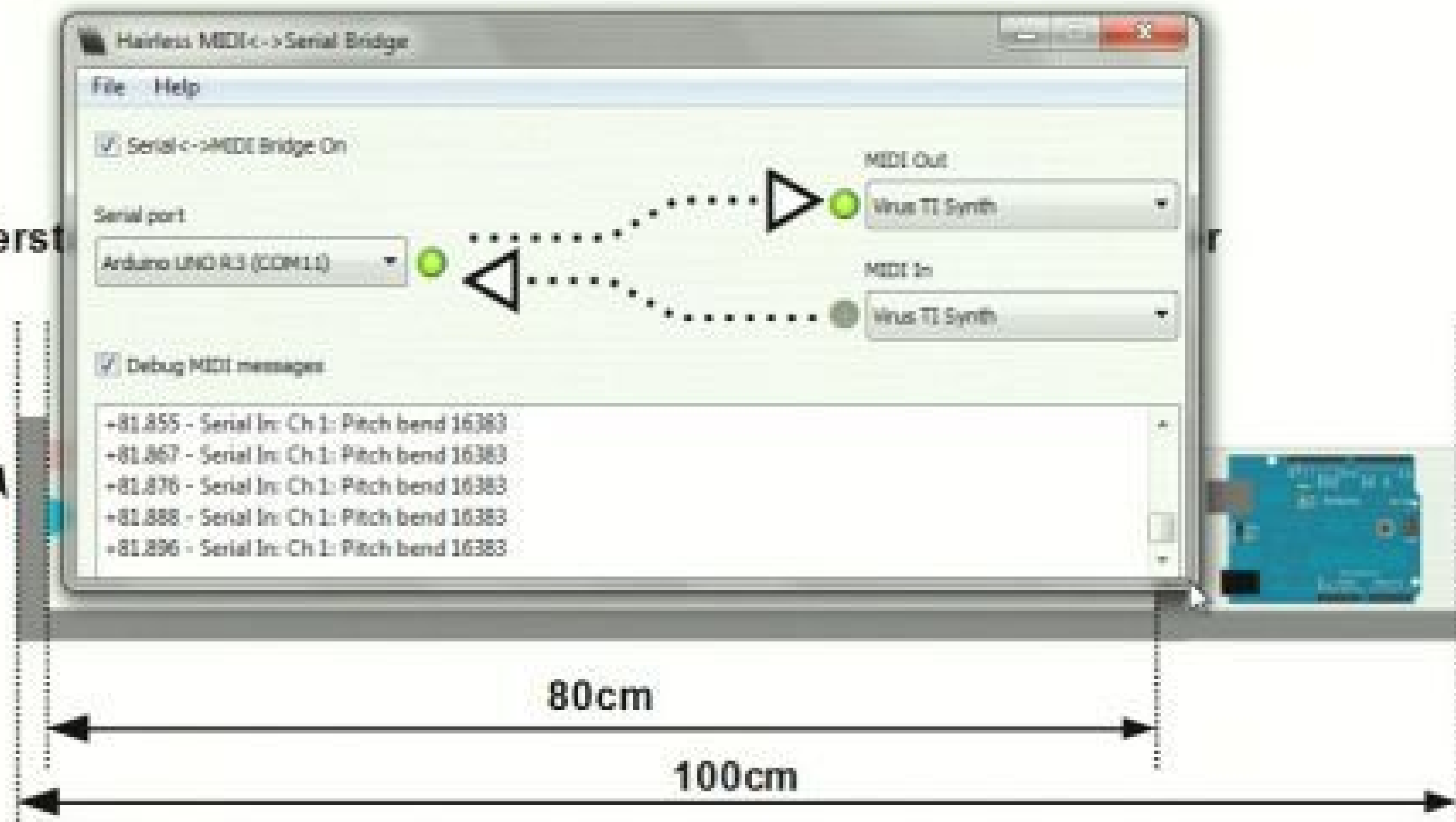
Aufbau

Fotowiderst



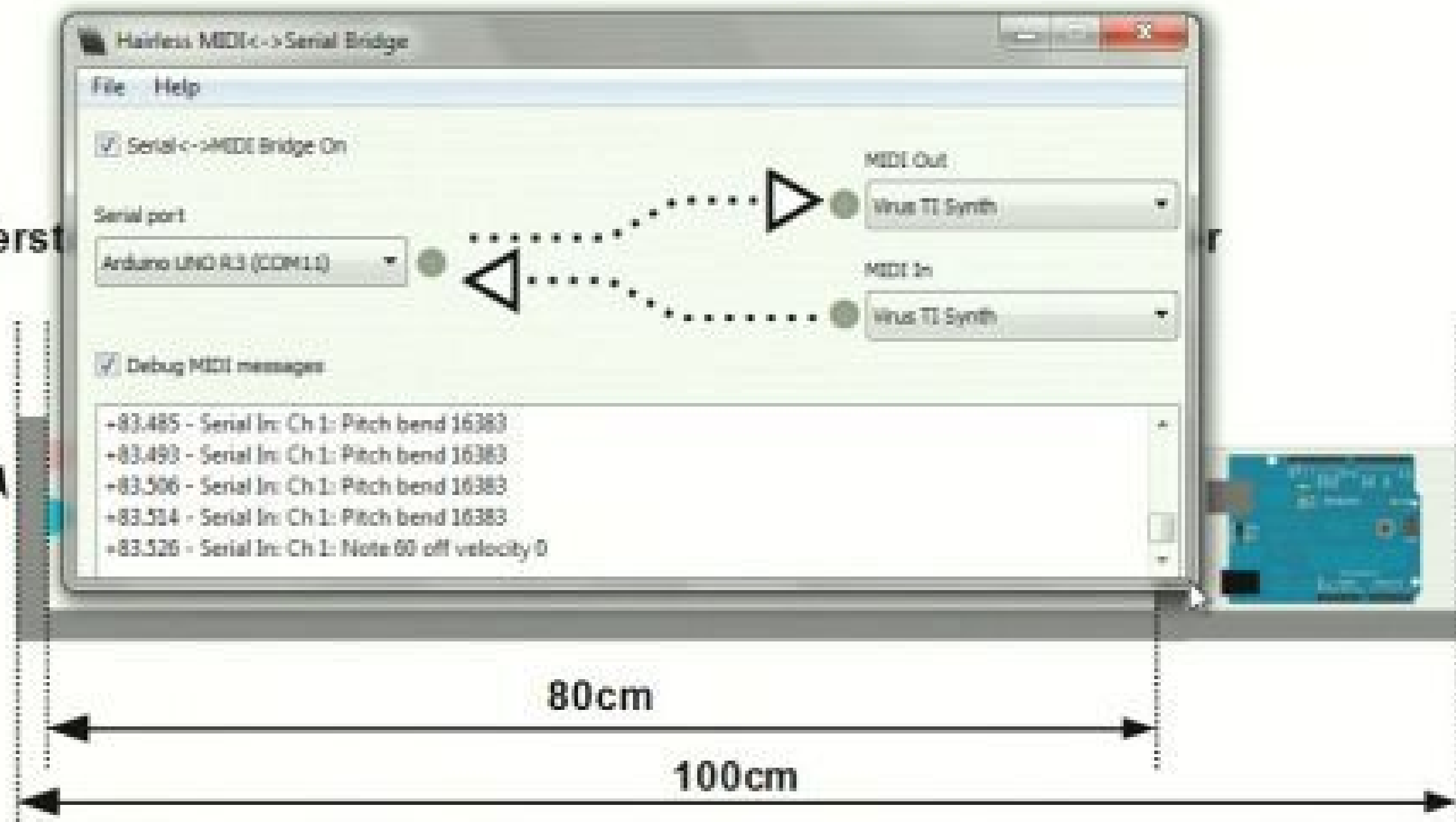
Aufbau

Fotowiderst



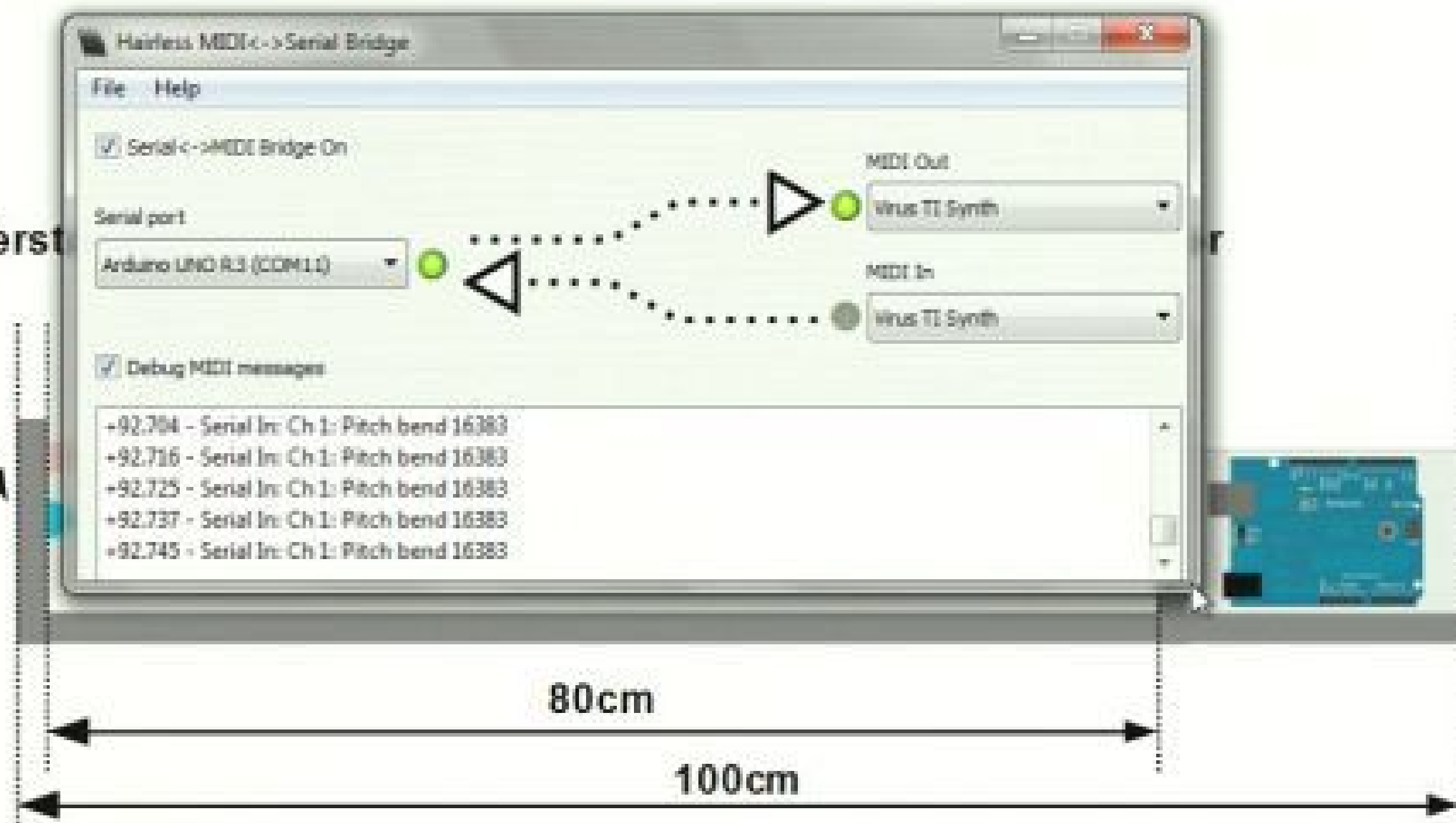
Aufbau

Fotowiderst



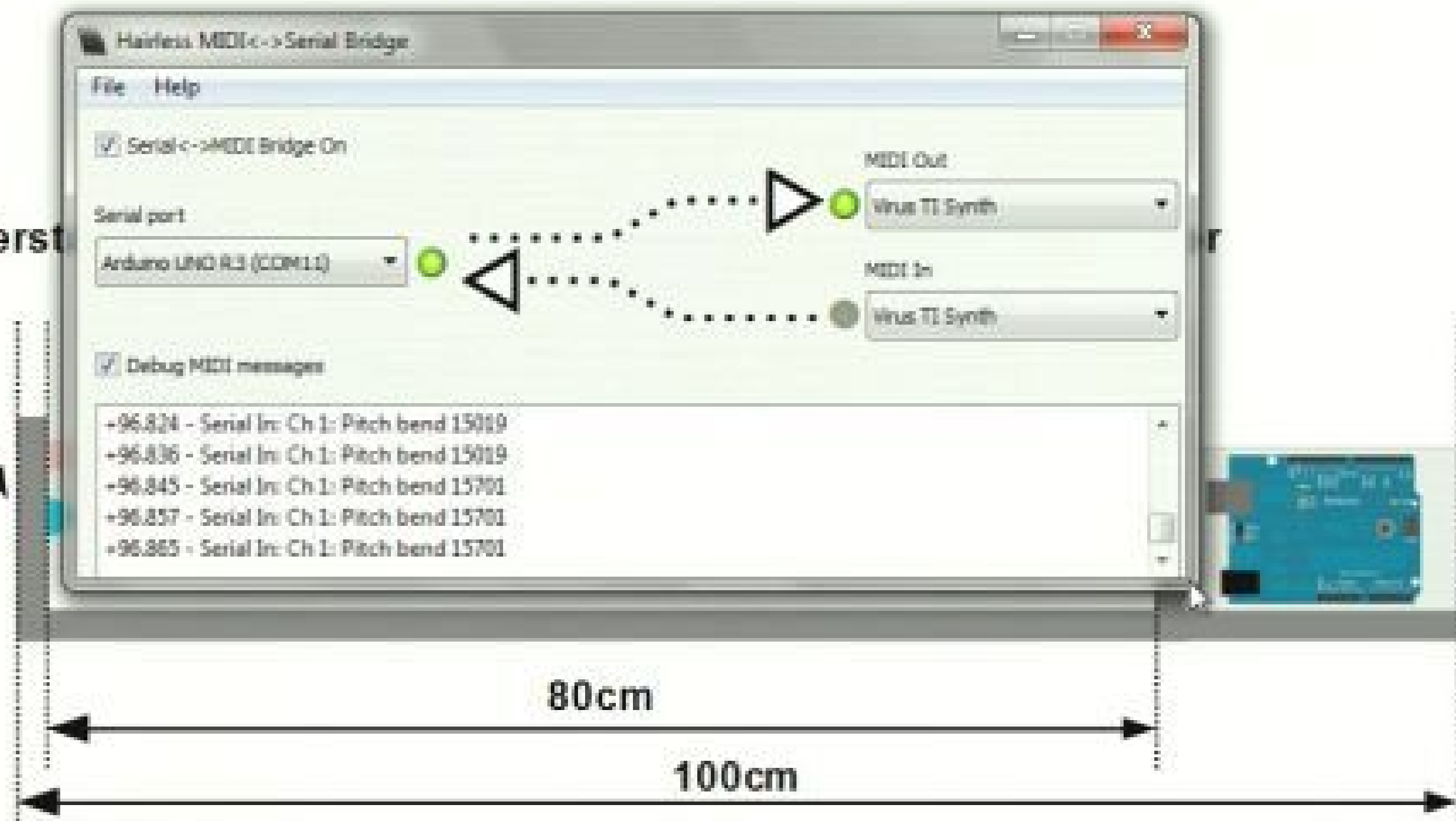
Aufbau

Fotowiderst



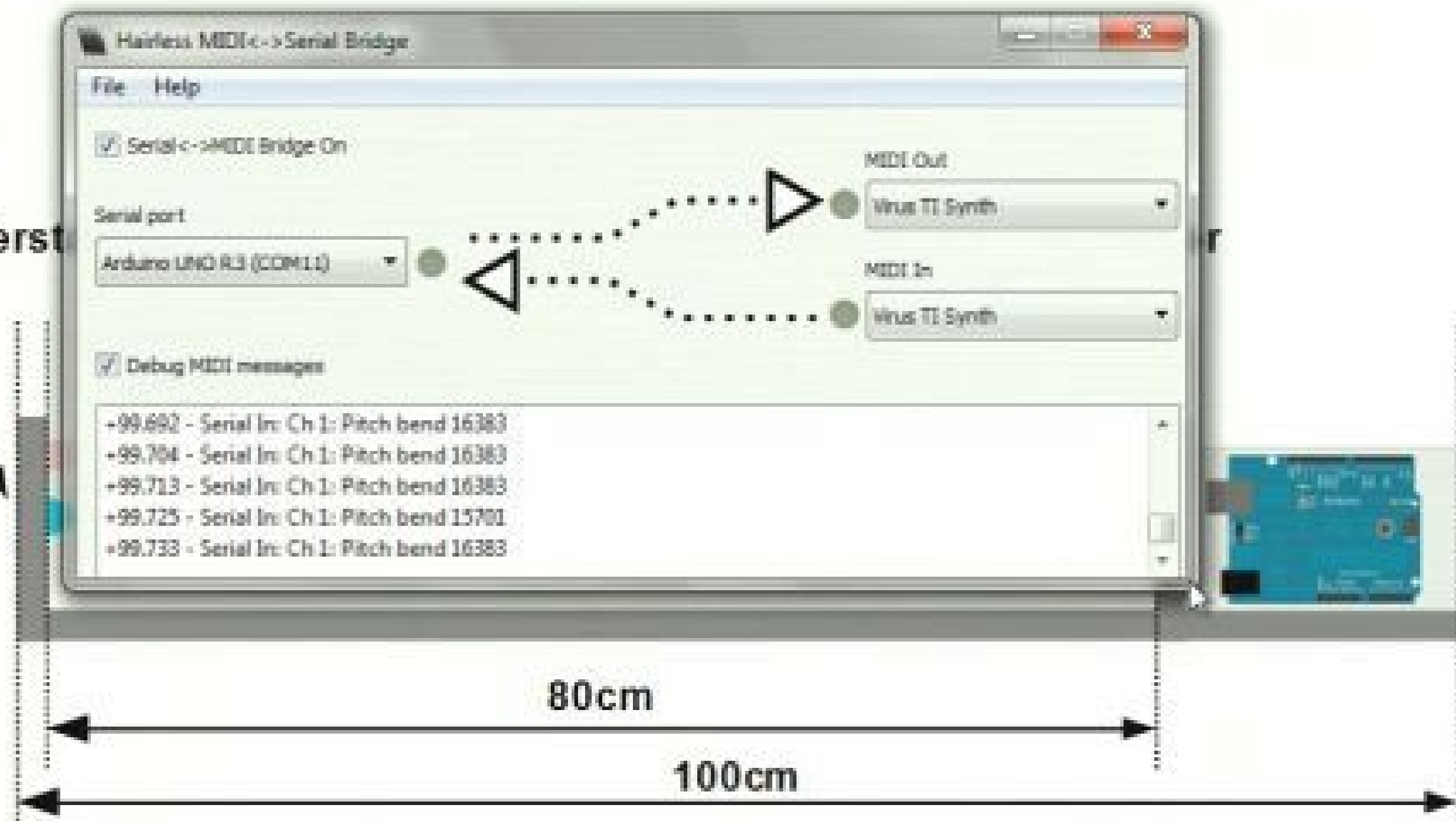
Aufbau

Fotowiderst



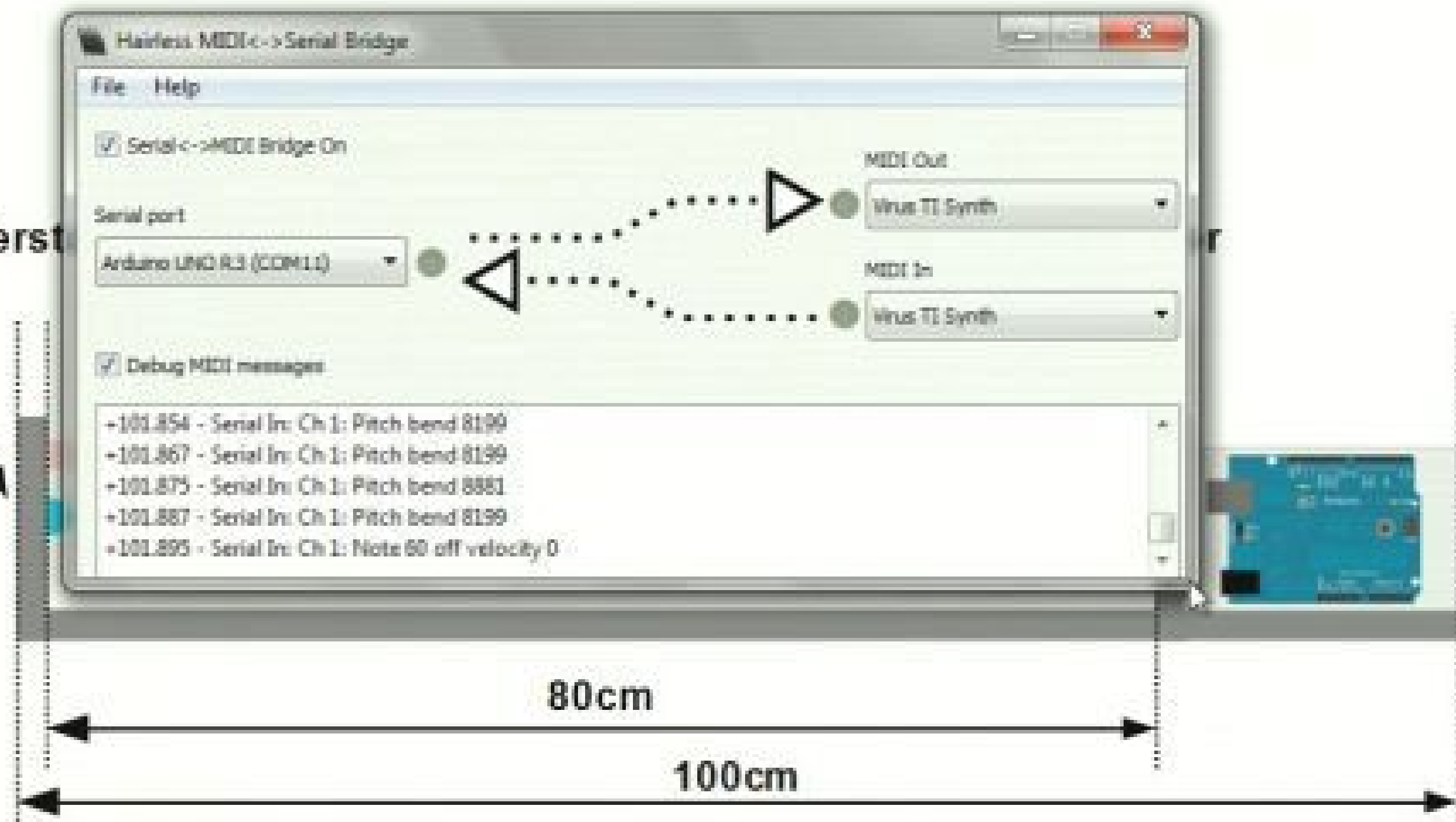
Aufbau

Fotowiderst



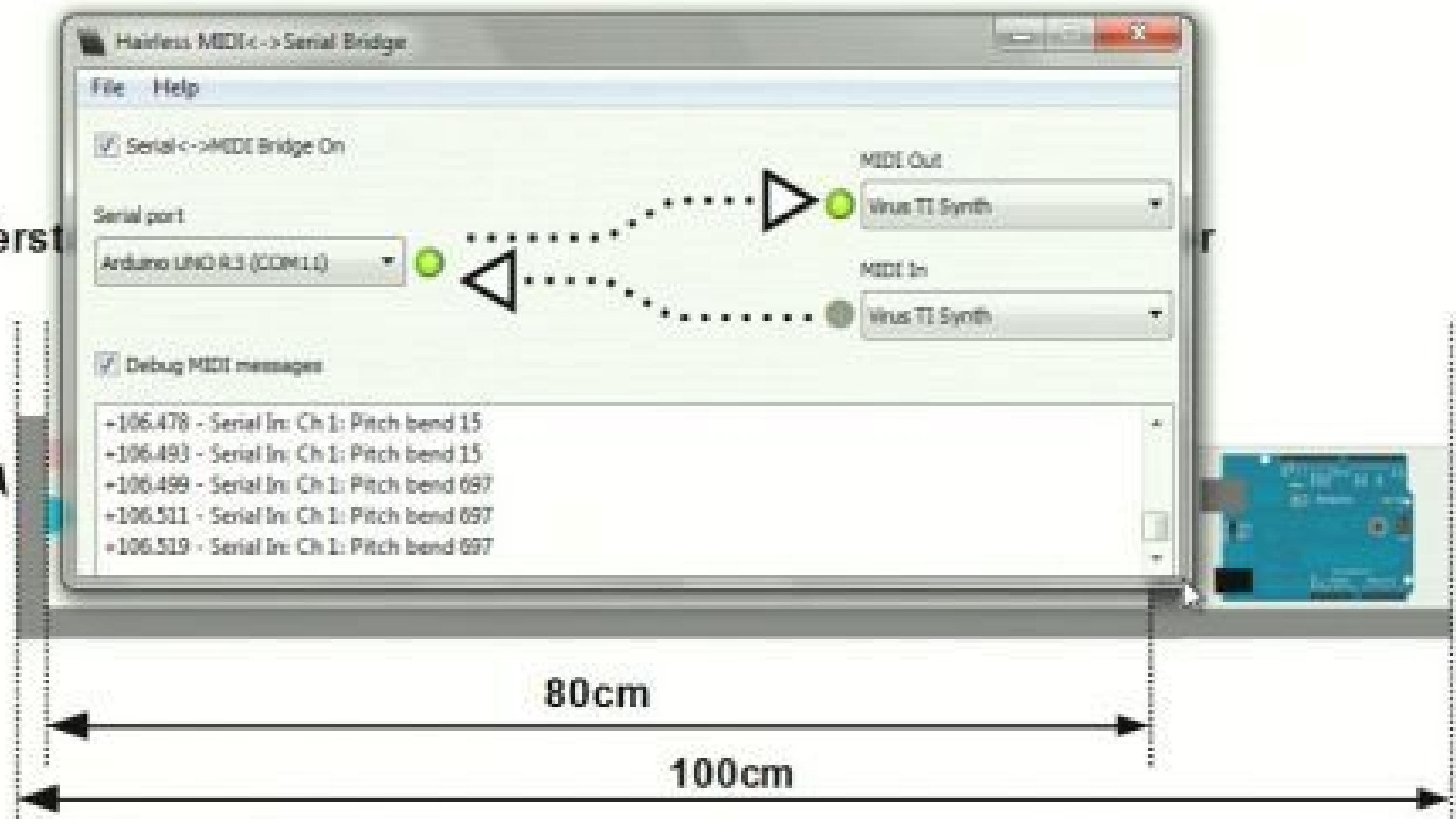
Aufbau

Fotowiderst



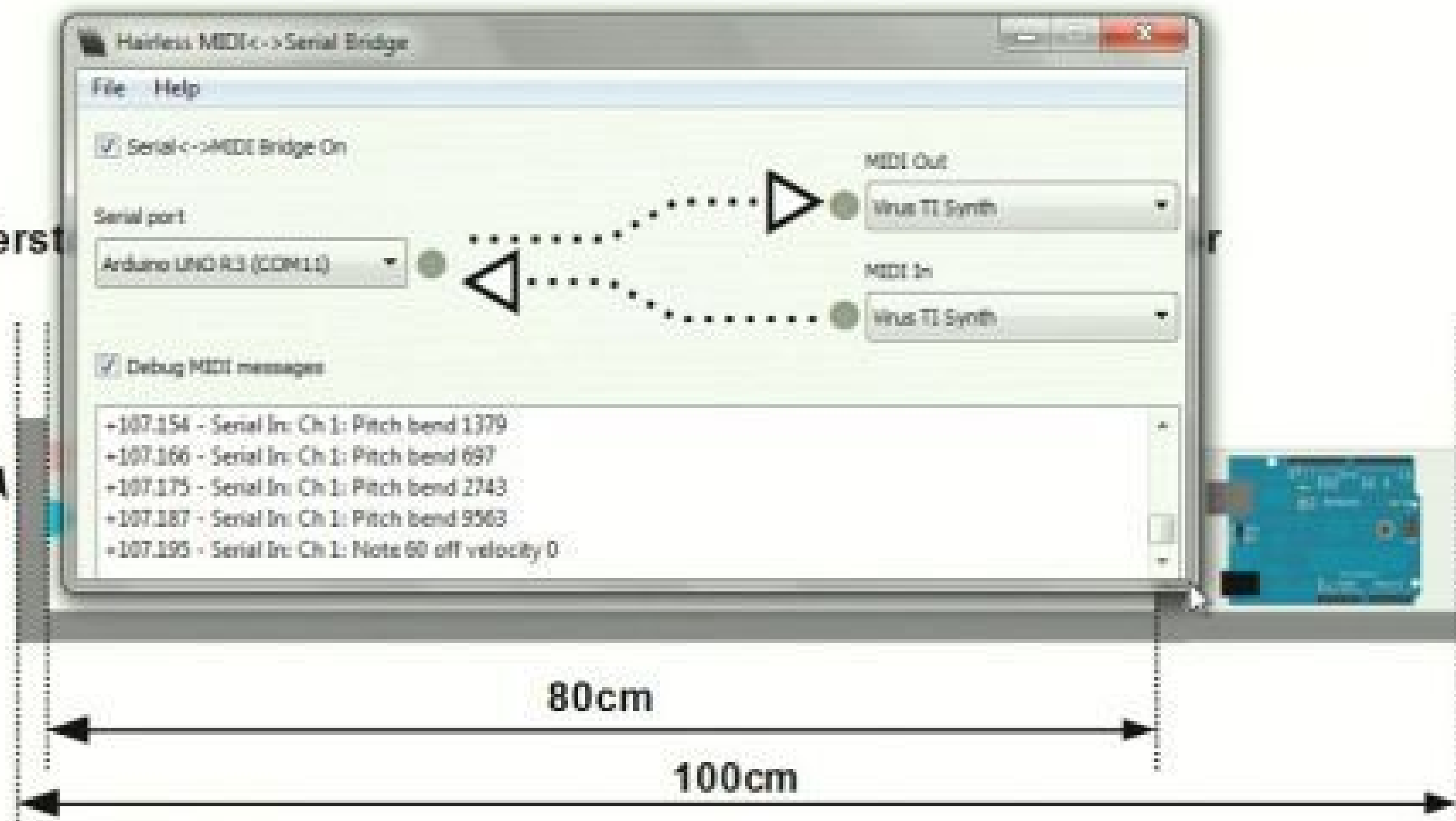
Aufbau

Fotowiderst

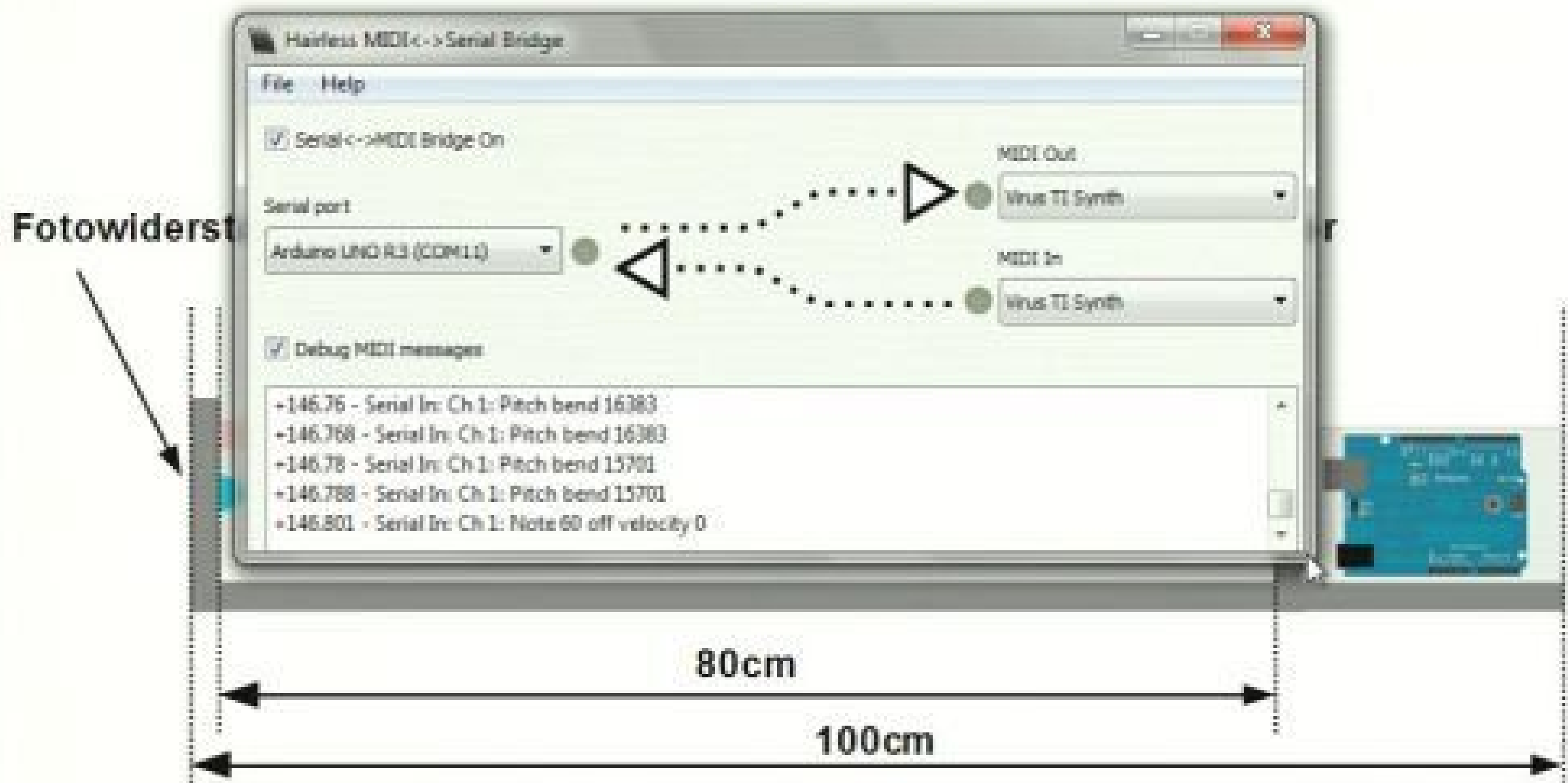


Aufbau

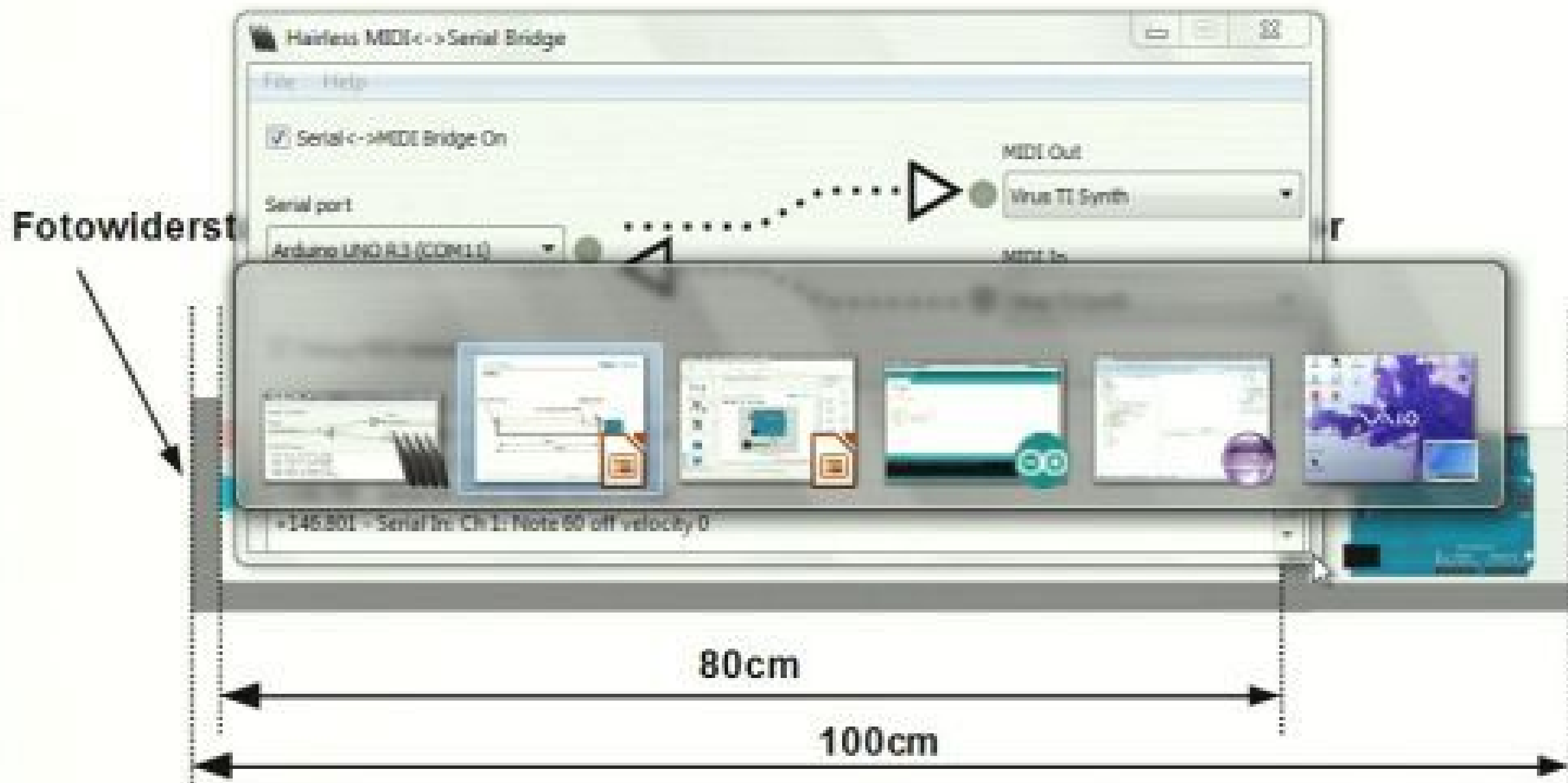
Fotowiderst



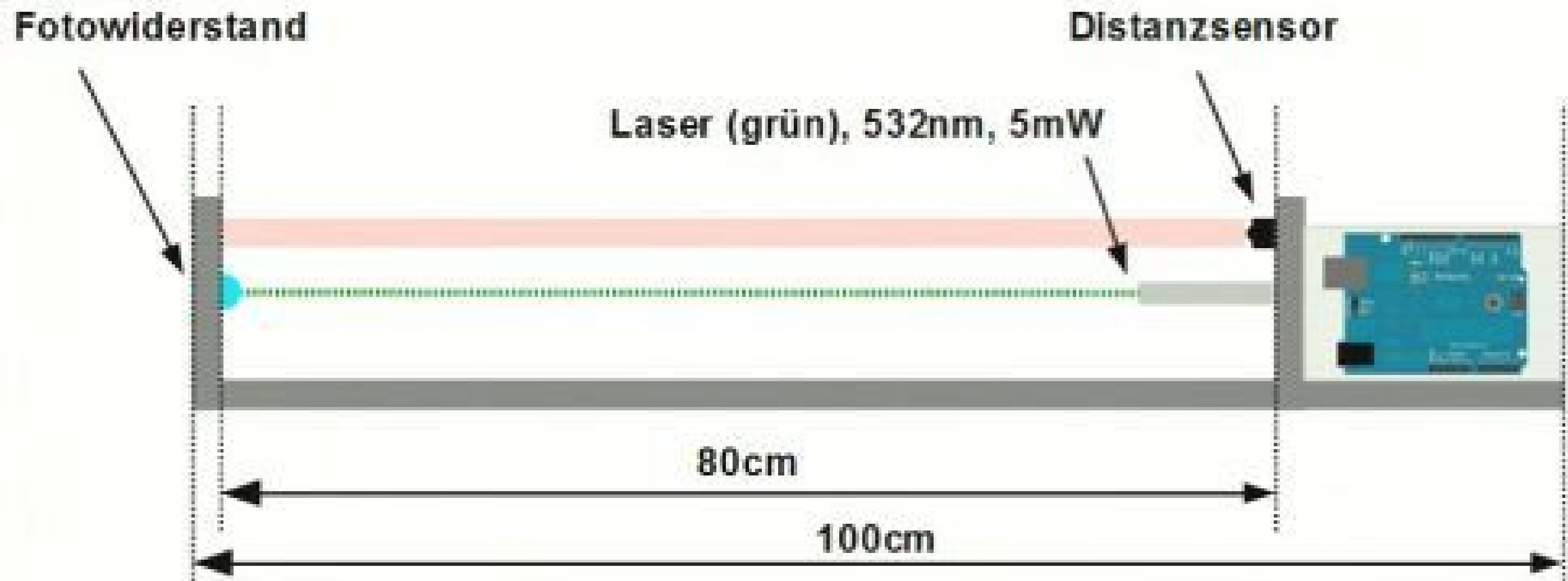
Aufbau



Aufbau



Aufbau

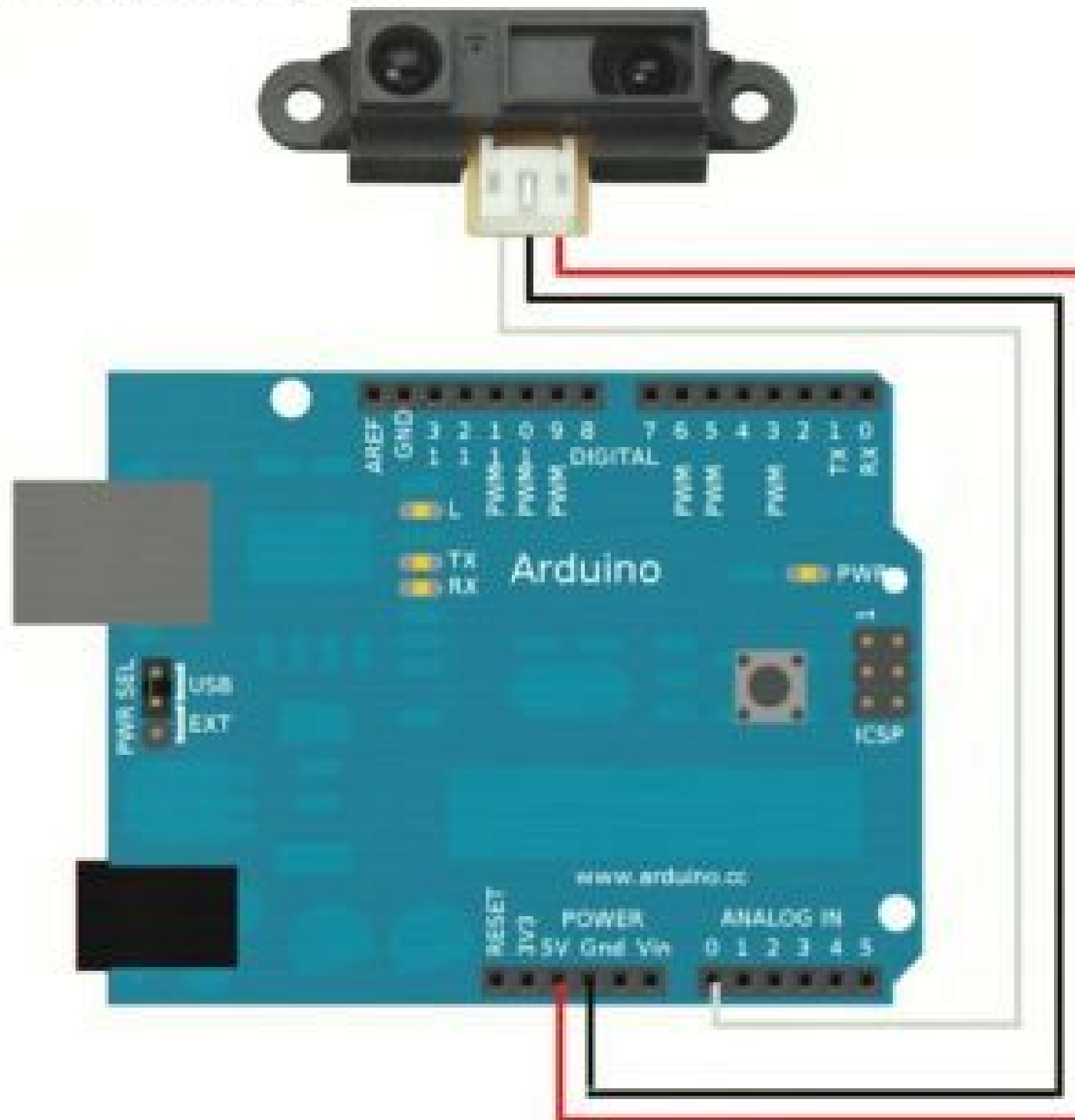


Distanzsensor Sharp GP2Y0A21YK0F

- Kombination eines PSD und IRED
 - PSD (Position Sensitive Detector)
 - IRED (Infrared emitting diode)
- Eigenschaften
 - 10cm bis 80cm Distanzmessung
 - Analog Output Type
 - Operating Voltage 4.5V – 5V
- Datenblatt auf http://www.sharpsma.com/webfm_send/1489



Distanzsensor anschliessen



Analoge Distanzsensordaten verarbeiten

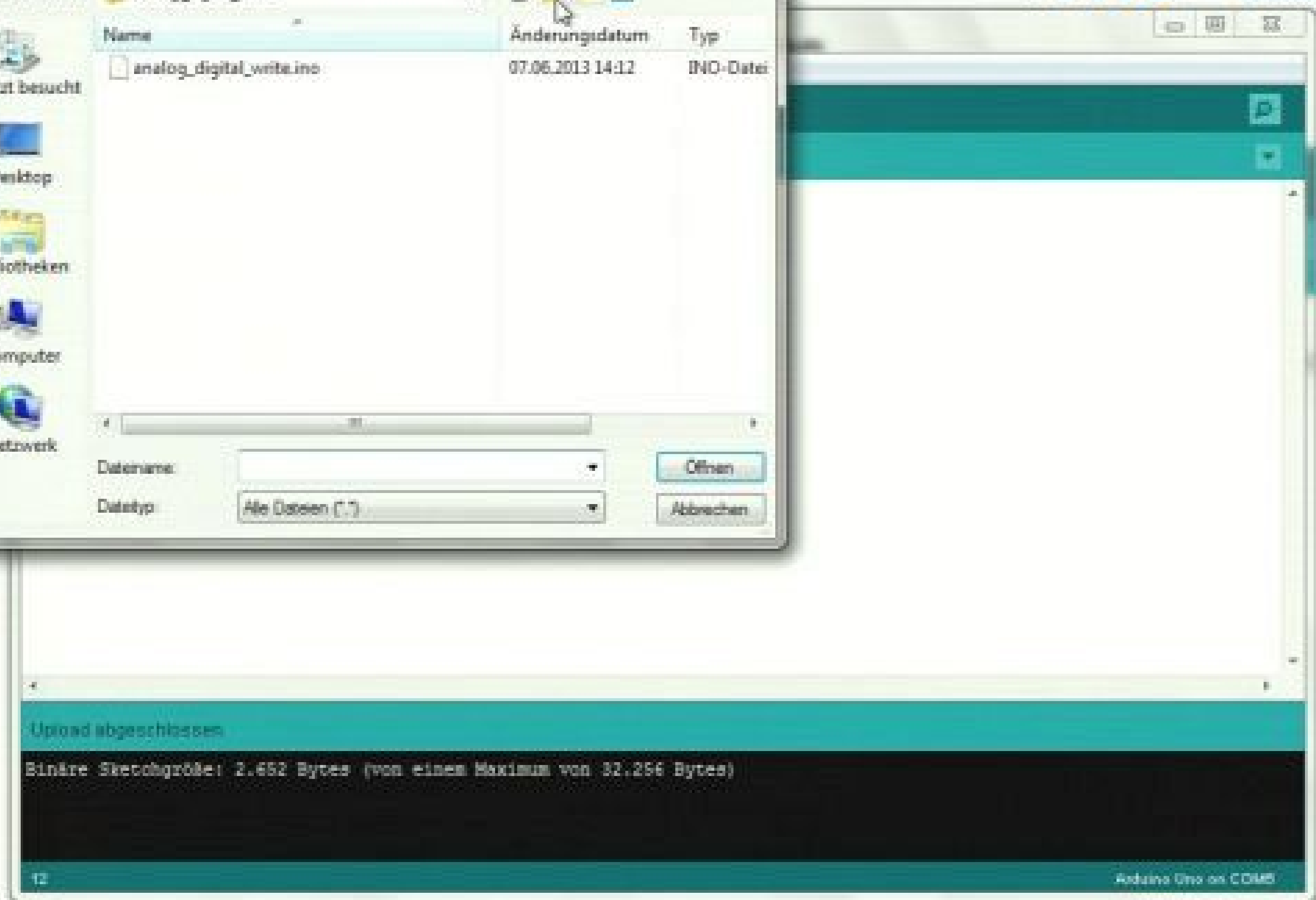


```
analog_in_GP2Y0A21YK0F
```

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```



pin

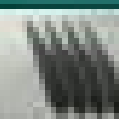


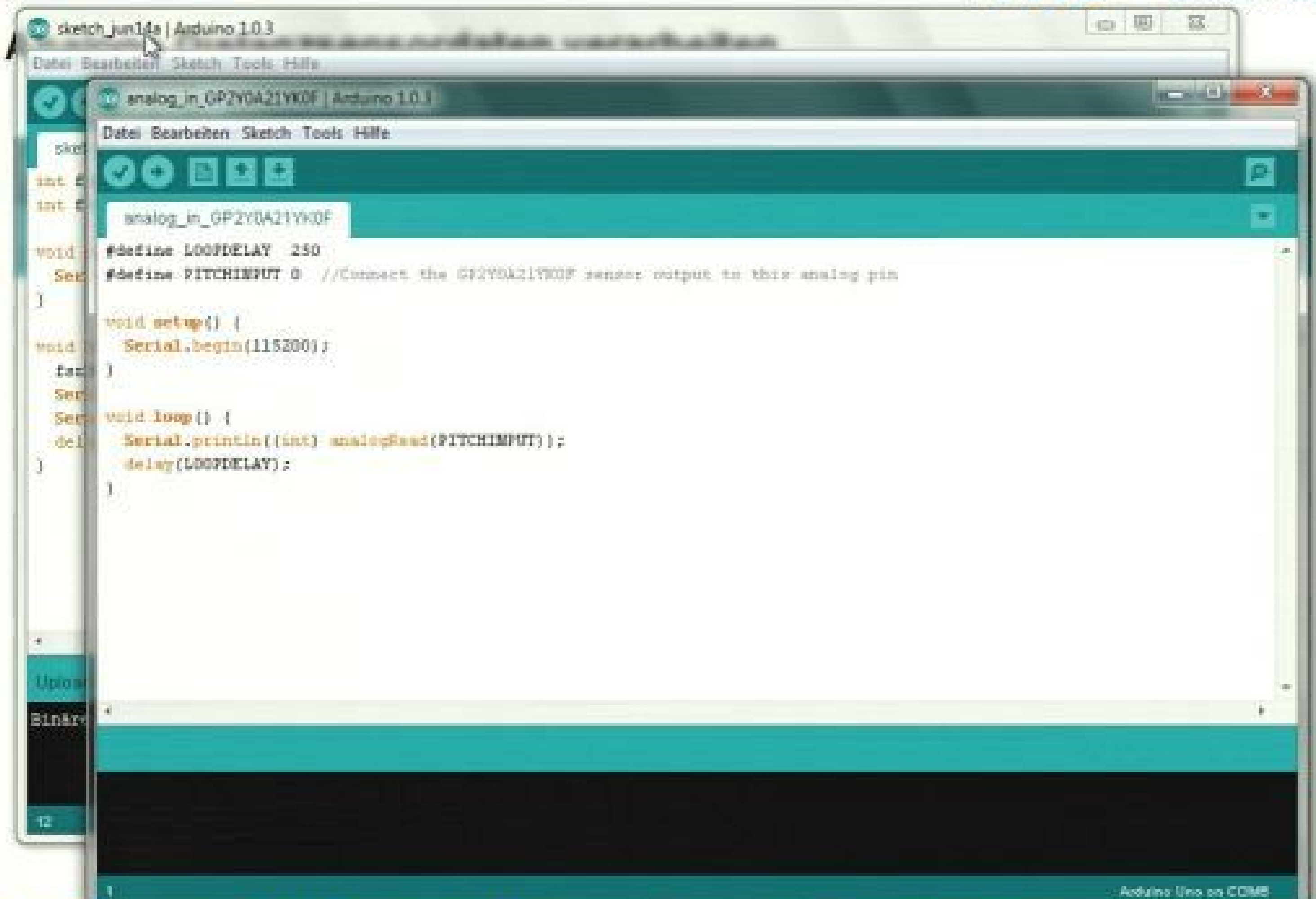
analog_in_GP2Y0A21YK0F

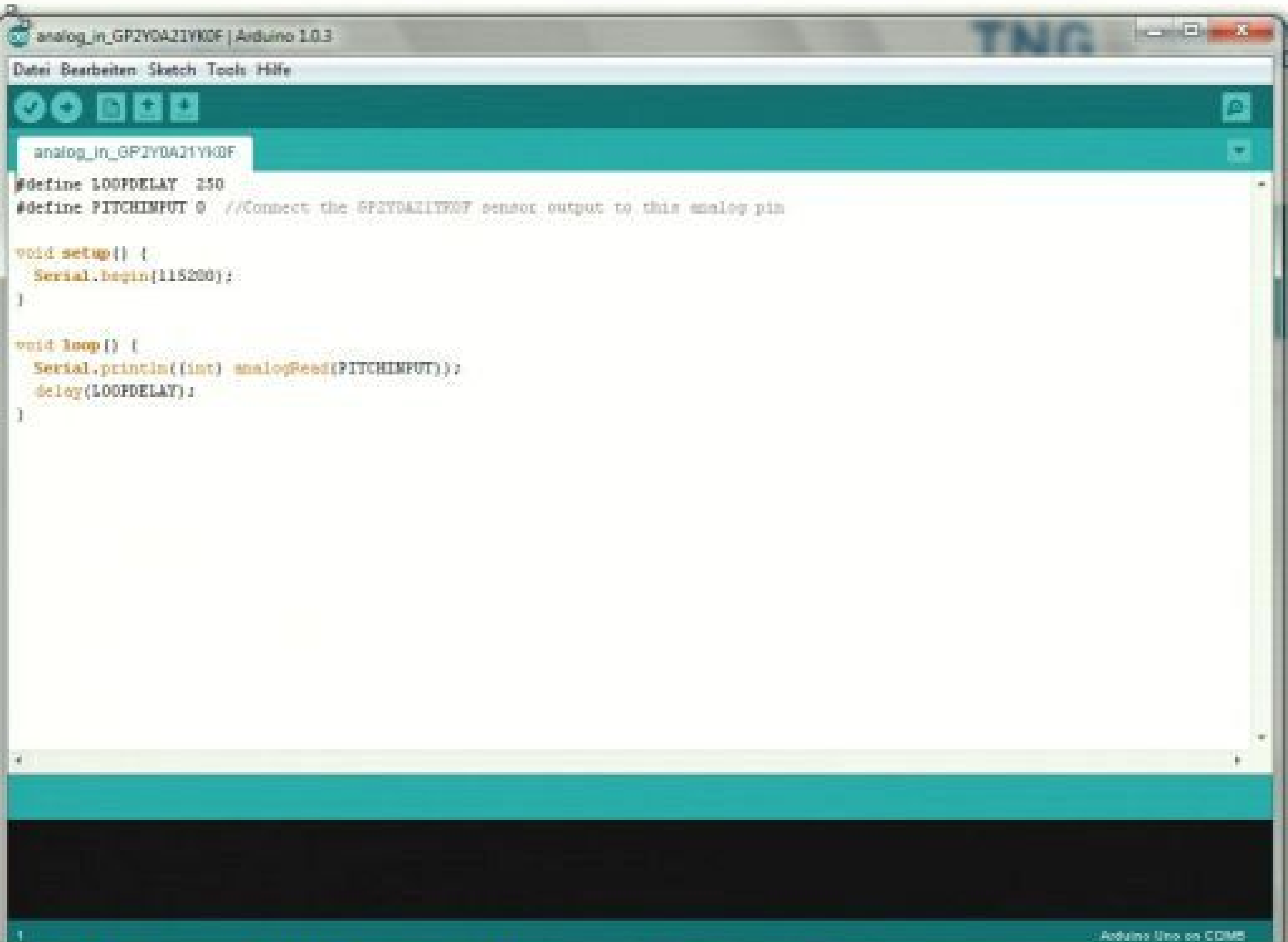
```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin
```

```
void setup() {
  Serial.begin(115200);
}
```

```
void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```







analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}

|
```



analog_in_GP2Y0A21YK0F

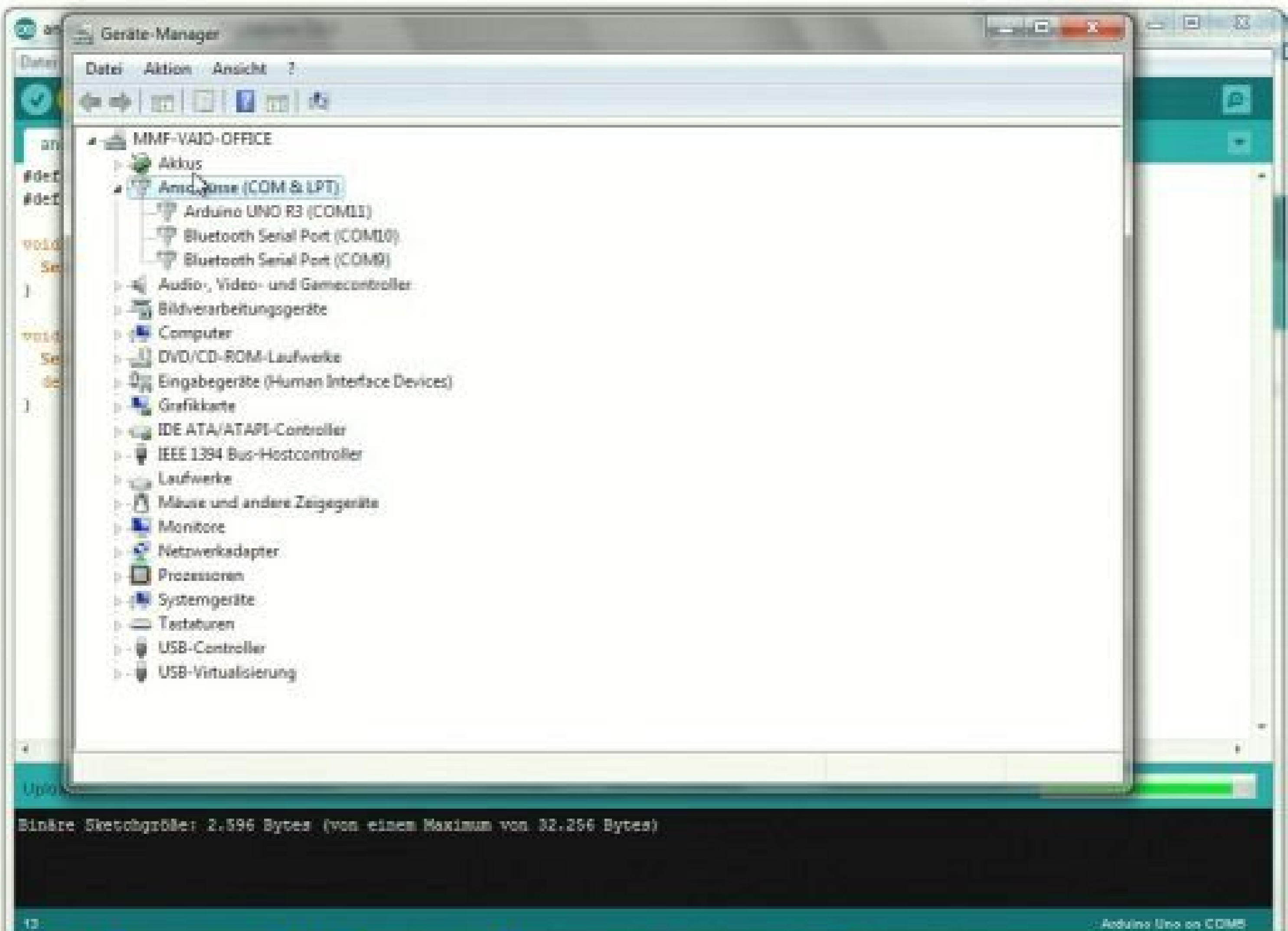
```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin

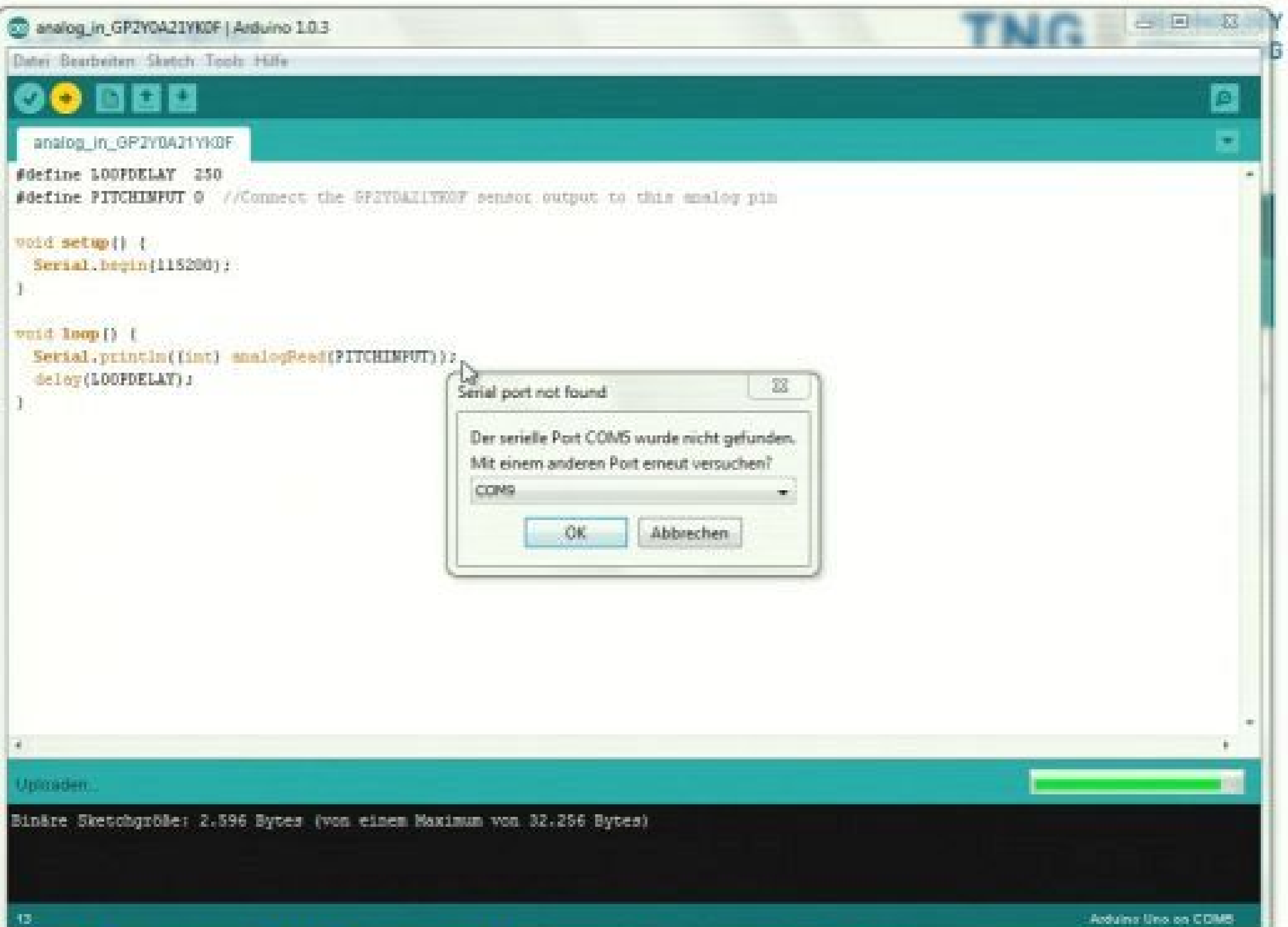
void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```

Sketch kompilieren...









analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```

Sketch kompilieren...



Binäre Sketchgröße: 2.596 Bytes (von einem Maximum von 32.256 Bytes)





analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}

|
```

Der serielle Port 'COM11' wird bereits verwendet. Probieren Sie, andere Programme zu beenden, die ihn benutzen könnten.

```
at: processing.app.Editor$DefaultExportHandler.run(Editor.java:2280)
at: processing.app.Editor$DefaultExportHandler.run(Editor.java:2280)
at: java.lang.Thread.run(Thread.java:619)
```




analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}

|
```

Sketch kompilieren...





analog_in_GP2Y0A21YK0F

#define LOOPDELAY 2

#define PITCHINPUT 0

void setup() {

Serial.begin(115200);

}

void loop() {

Serial.println((int) analogRead(PITCHINPUT));

delay(LOOPDELAY);

}

Automatisch formatieren

Strg+T

Sketch archivieren

Kodierung reparieren & neu laden

Serial Monitor

Strg+Umschalt+M

Board

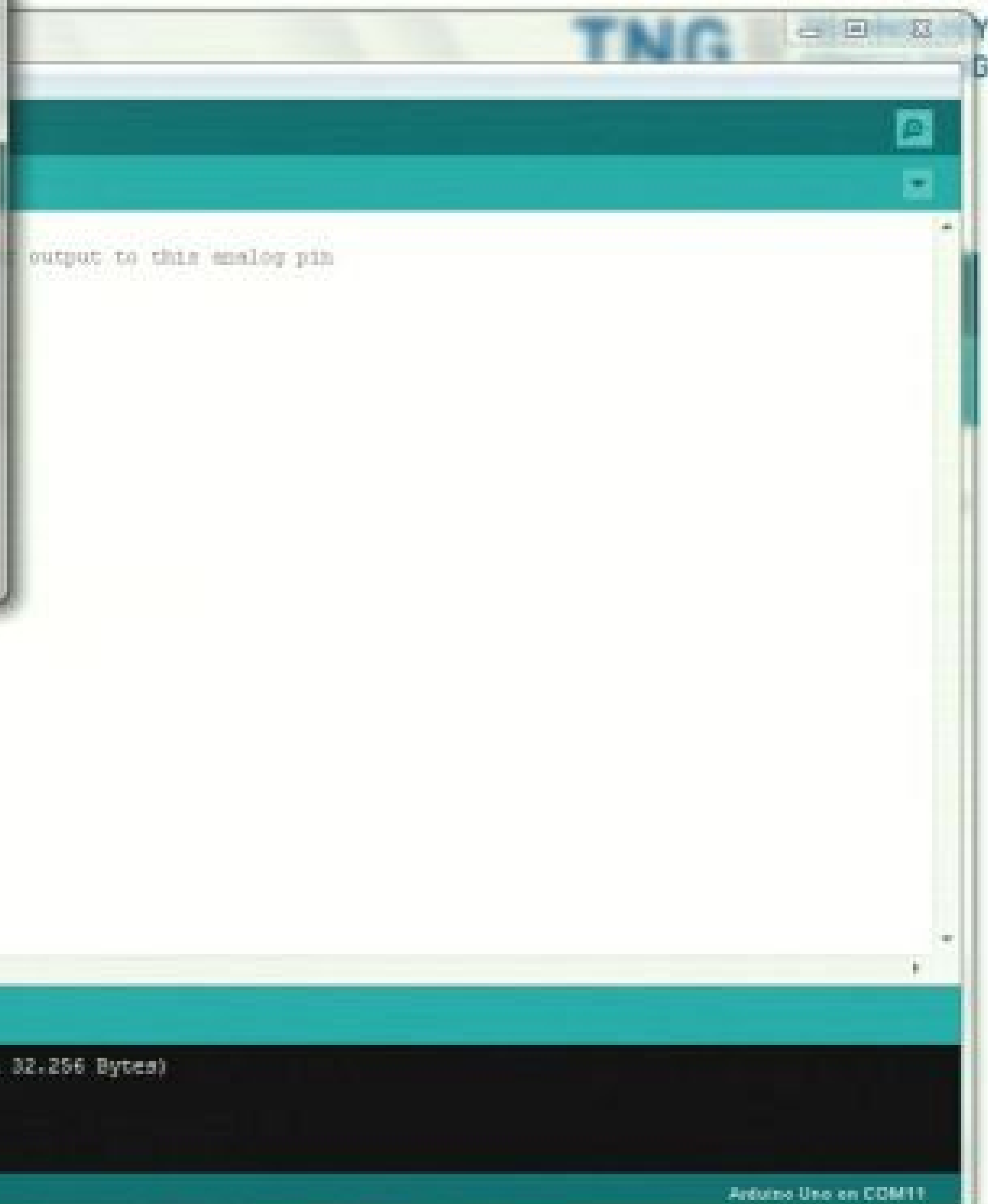
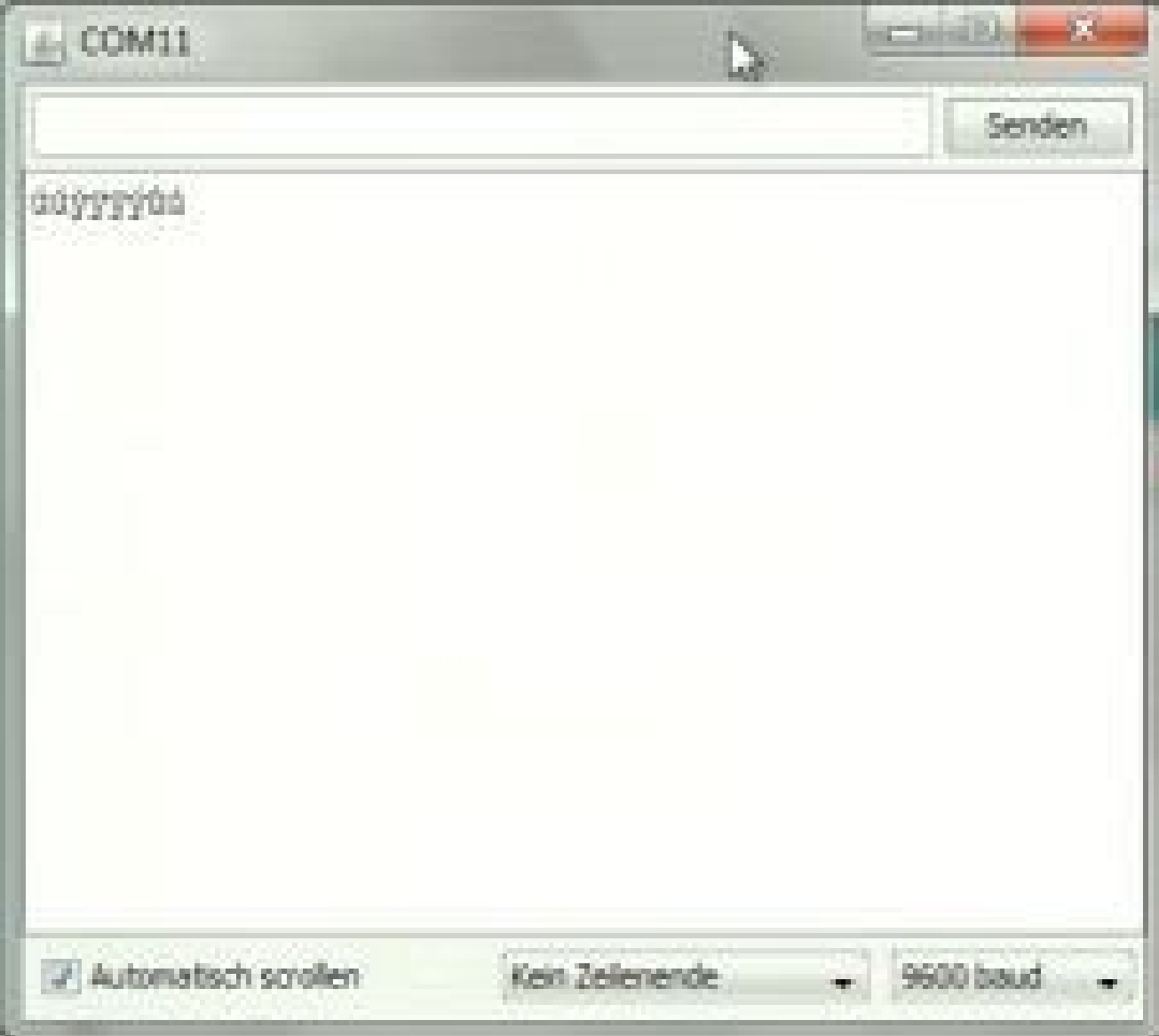
Serieller Port

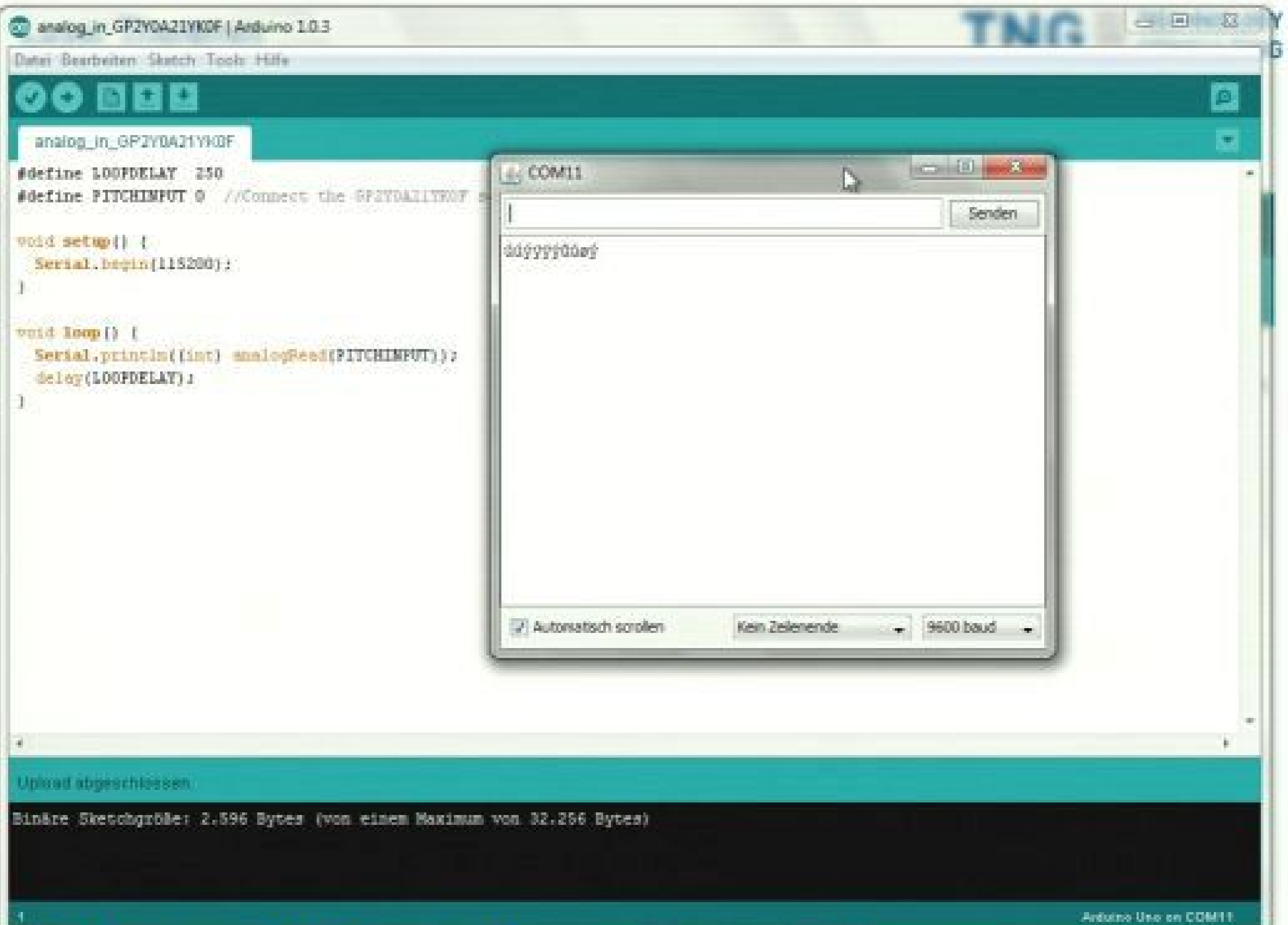
Programmer

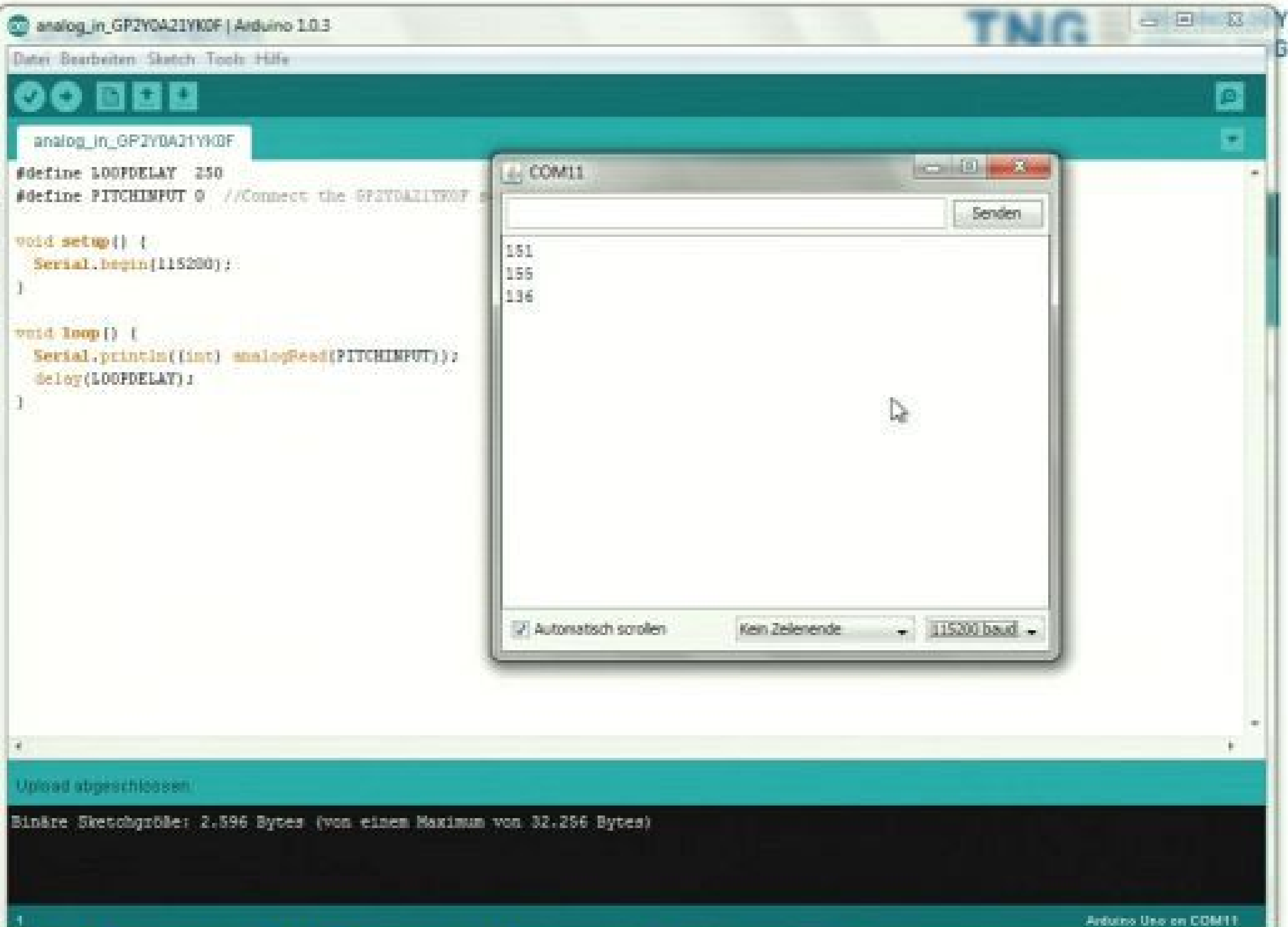
Bootloader installieren

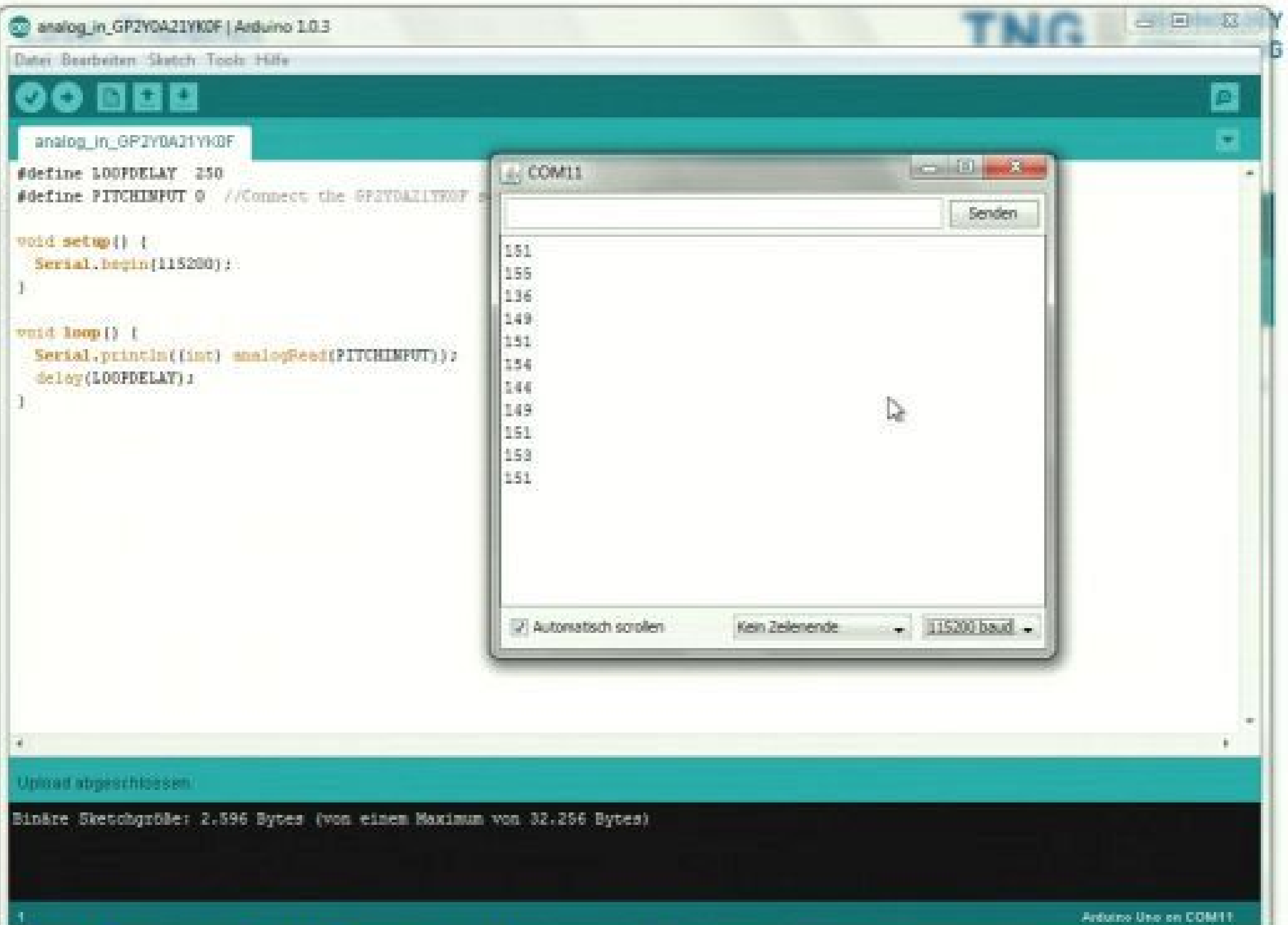
Uploade...

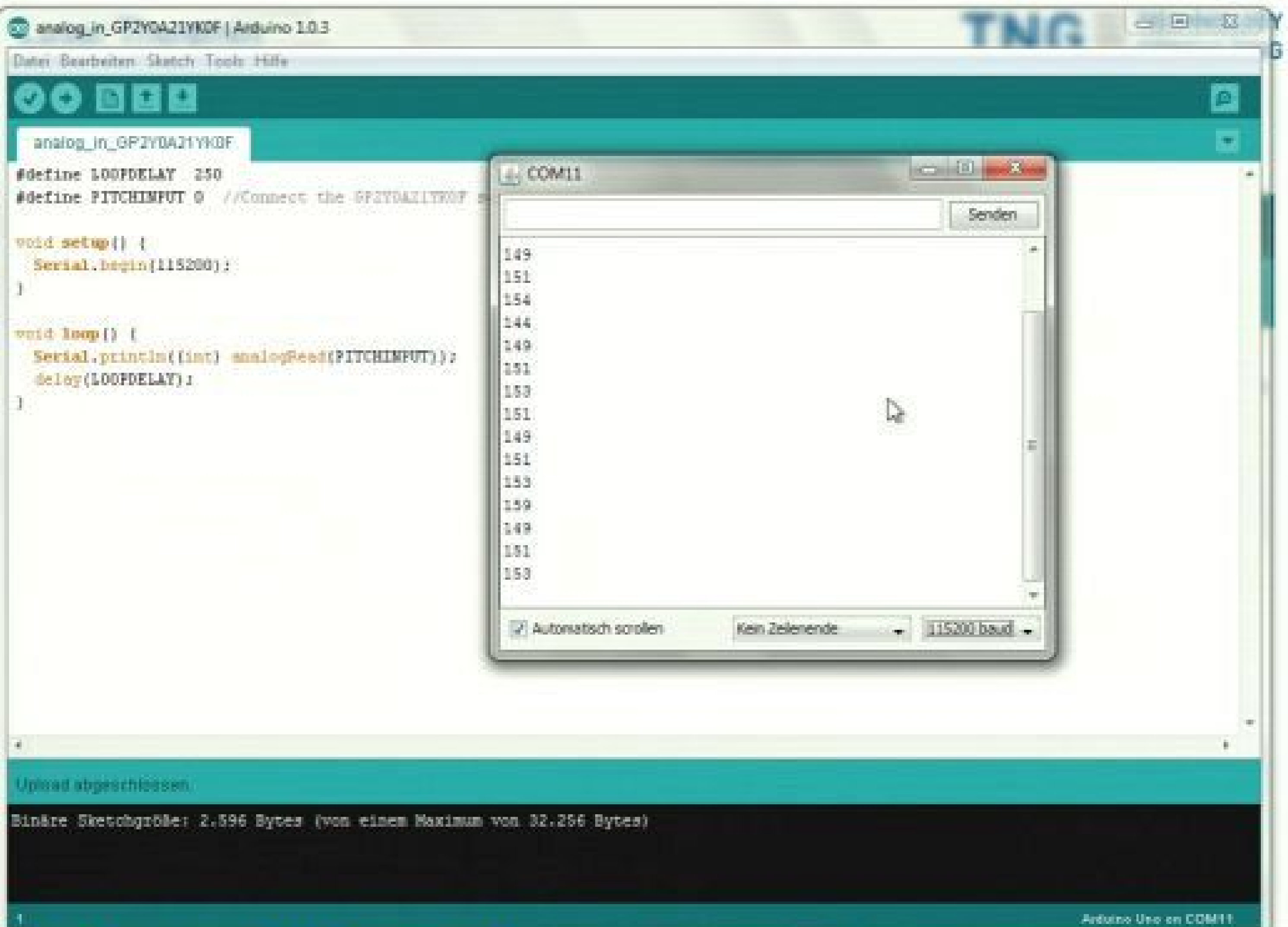
Binäre Sketchgröße: 2.596 Bytes (von einem Maximum von 32.256 Bytes)

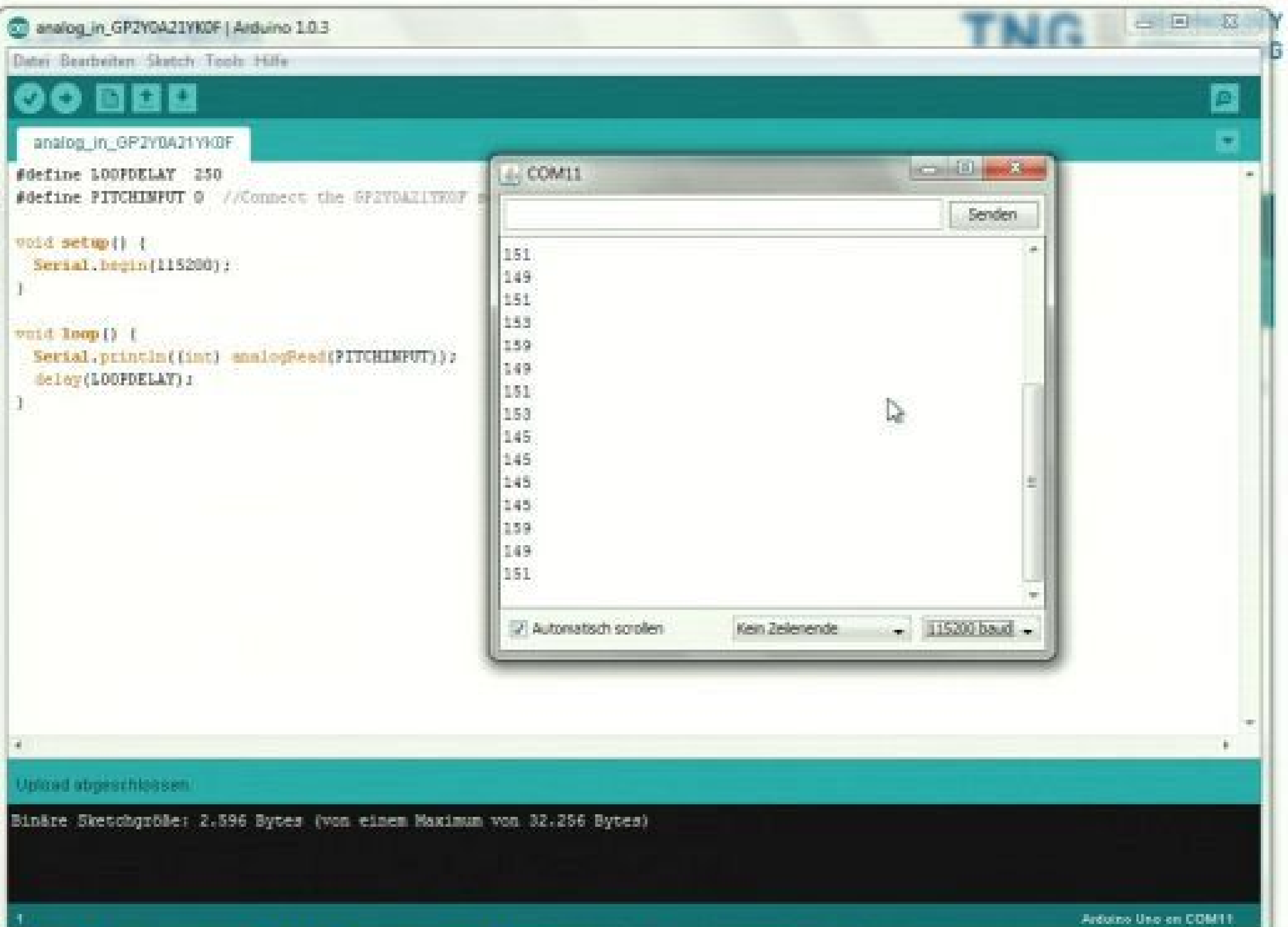












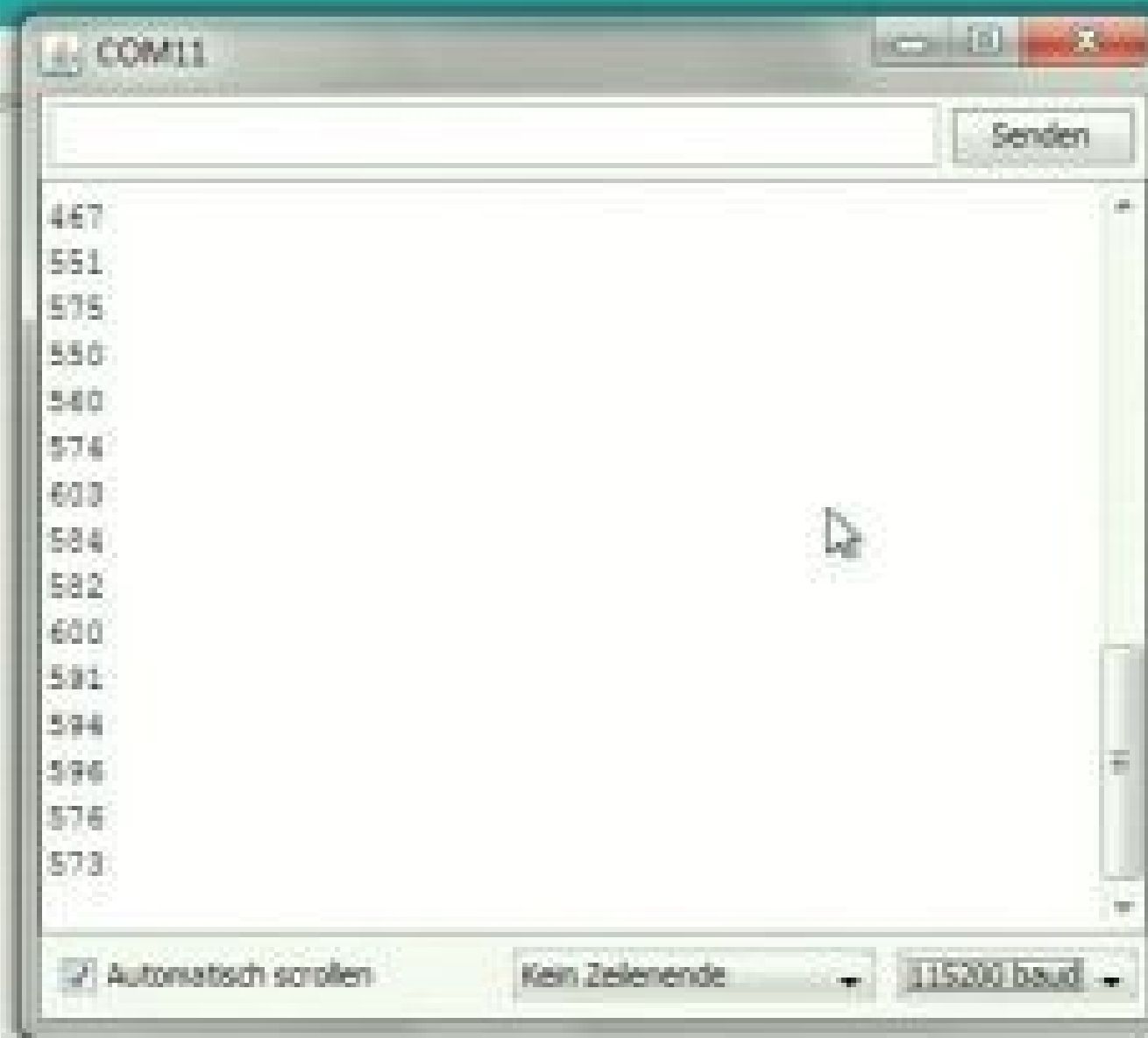


analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F to

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```



Upload abgeschlossen

Binäre Sketchgröße: 2.596 Bytes (von einem Maximum von 32.256 Bytes)

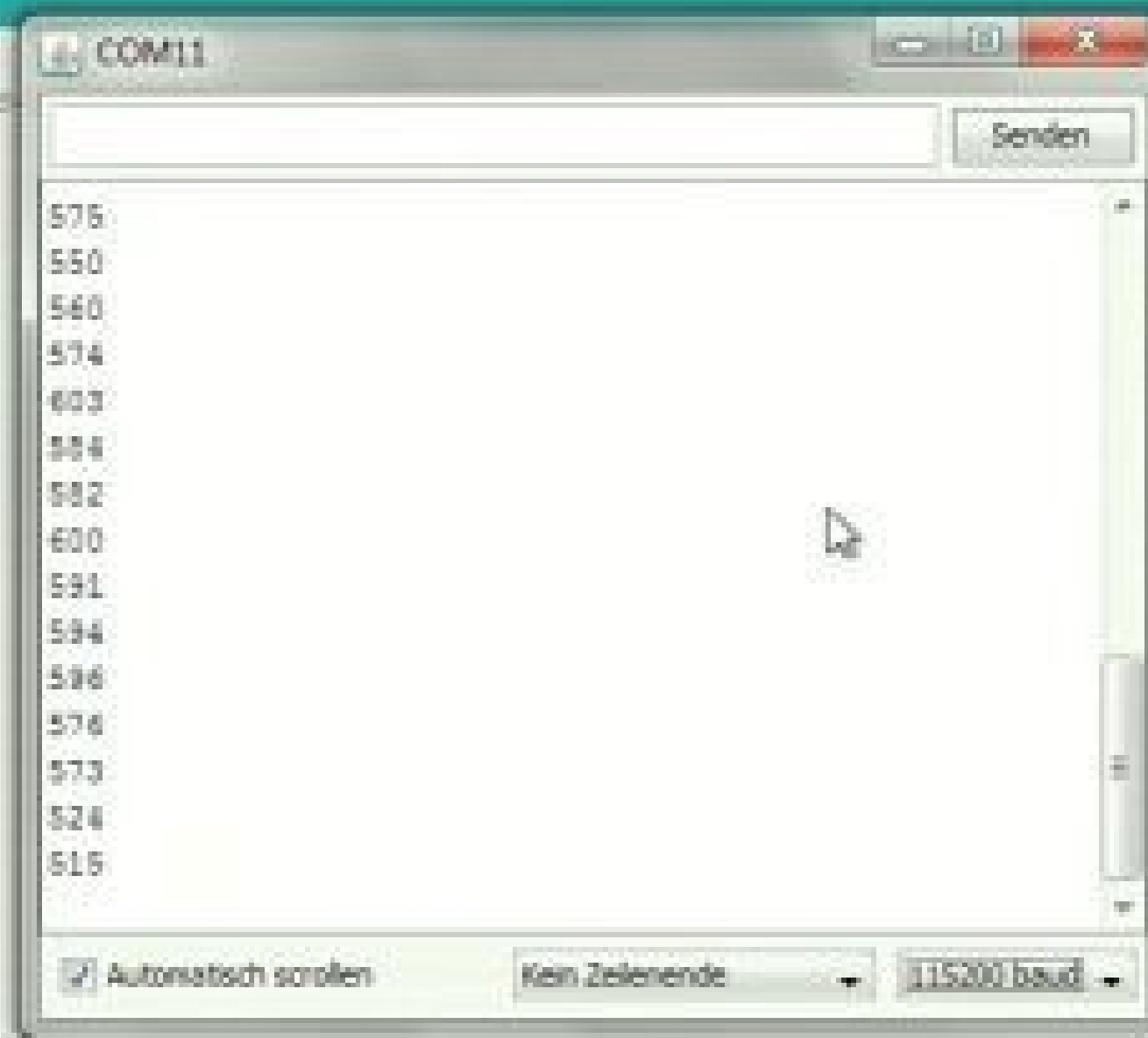


analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F to

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```



Upload abgeschlossen

Binäre Sketchgröße: 2.596 Bytes (von einem Maximum von 32.256 Bytes)

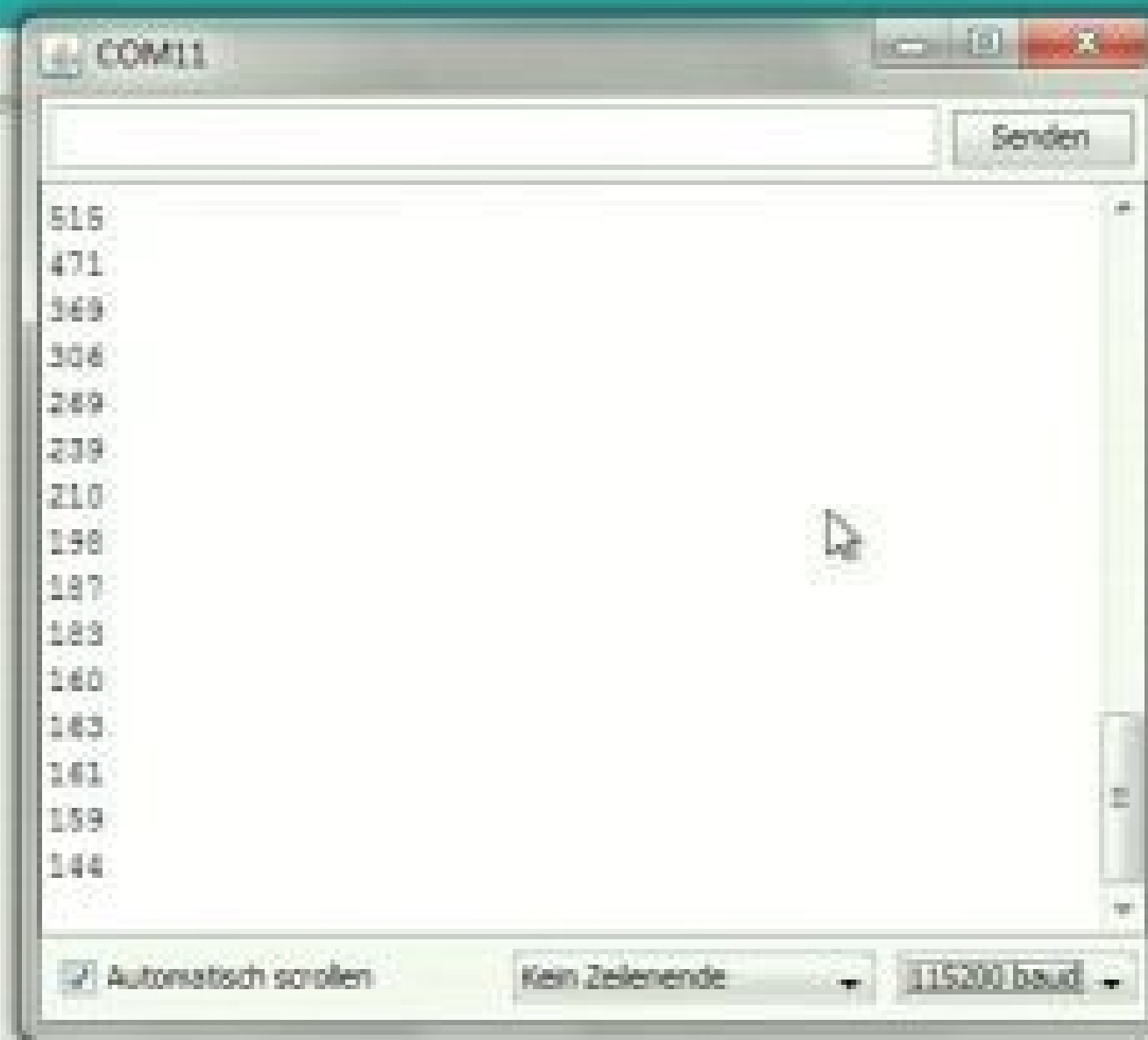


analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F to

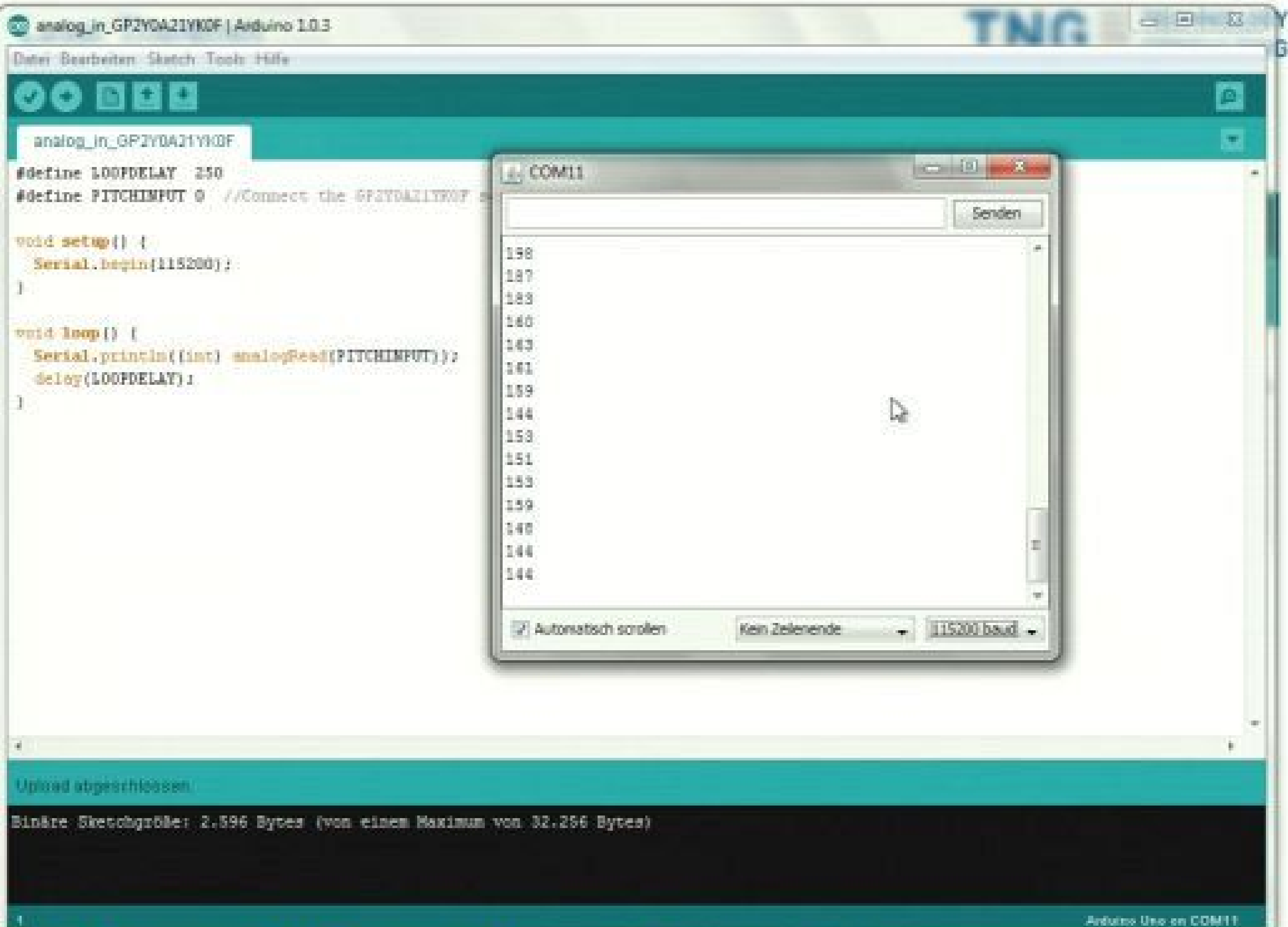
void setup() {
  Serial.begin(115200);
}

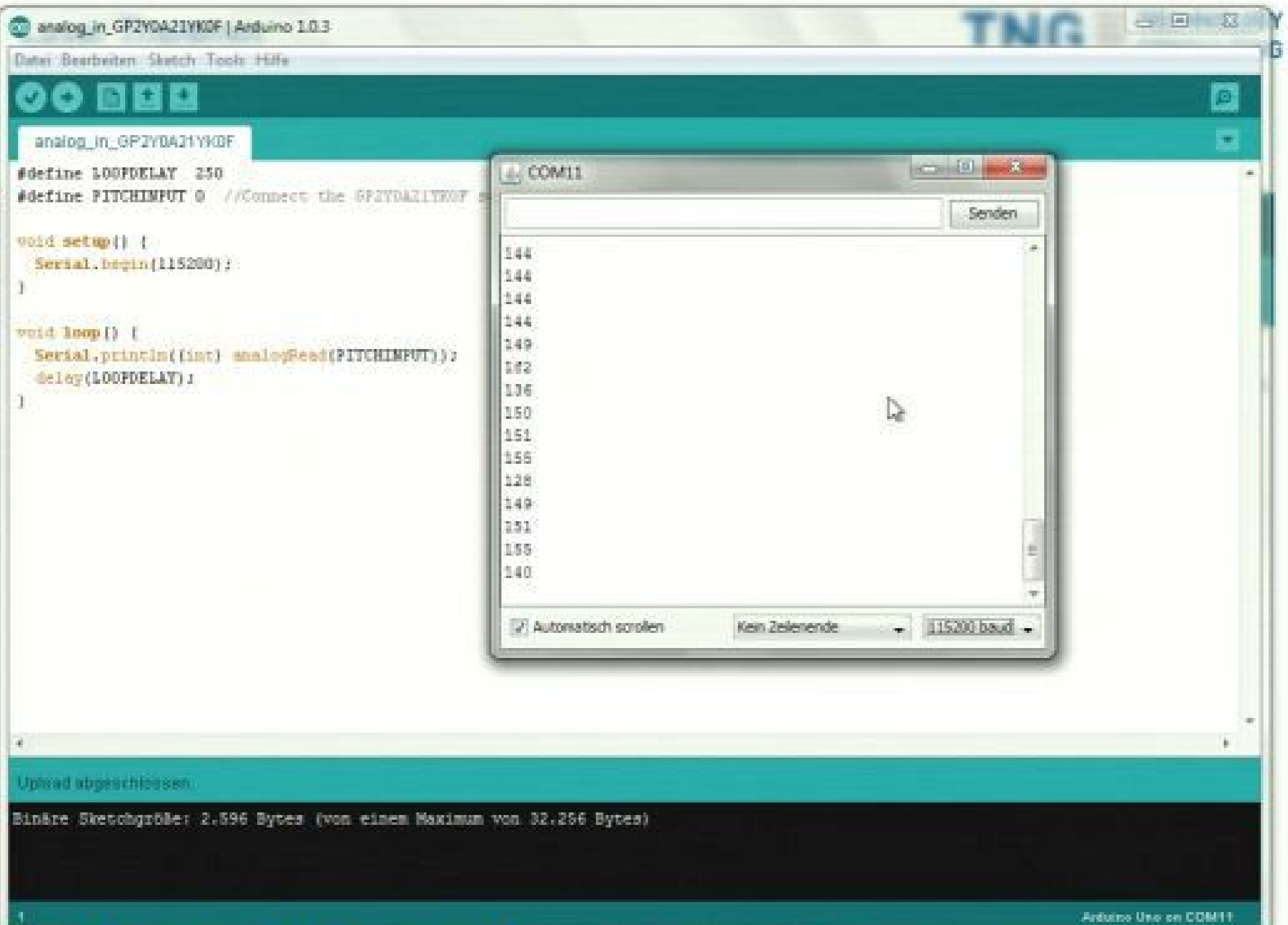
void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```

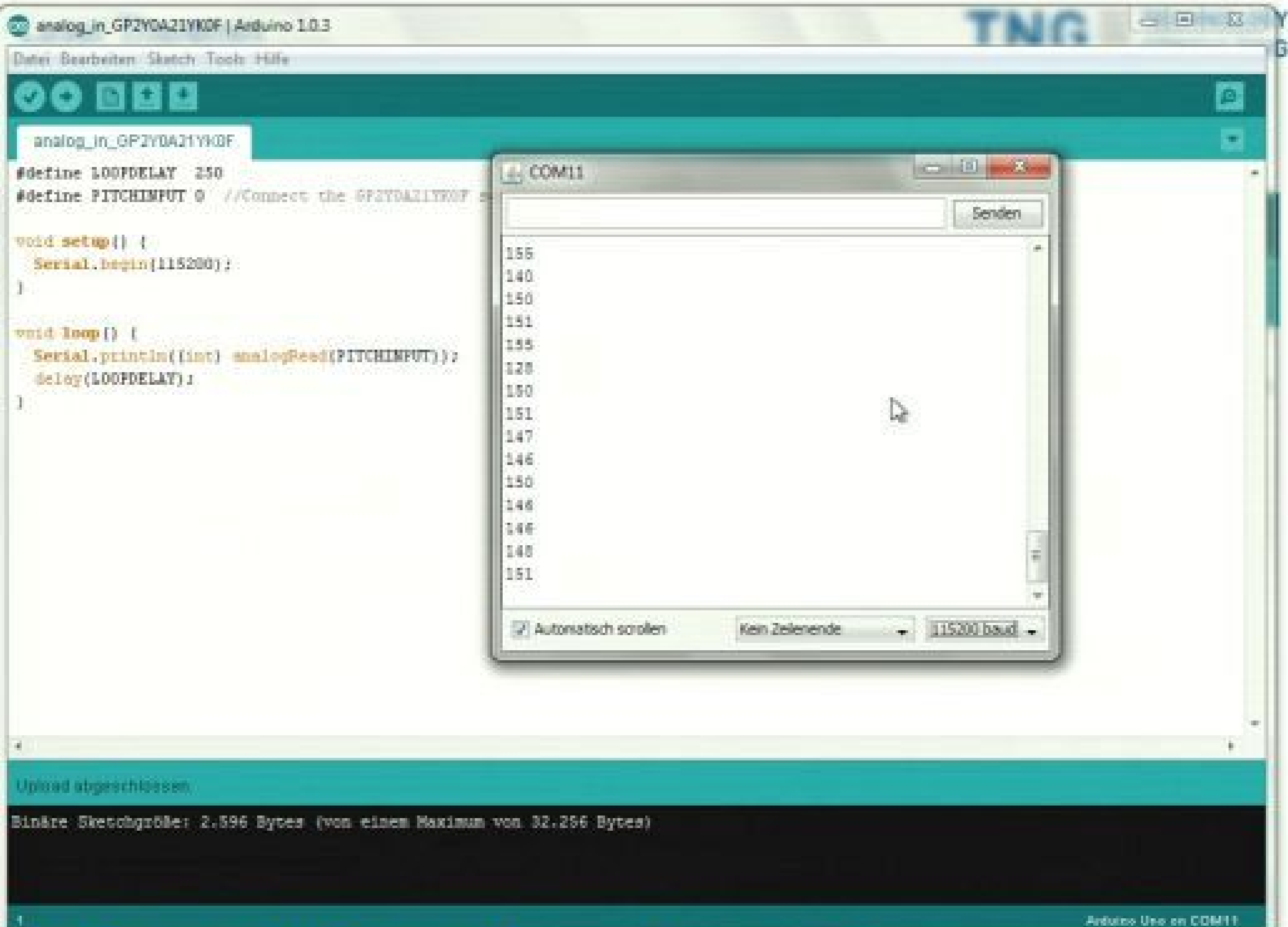


Upload abgeschlossen

Binäre Sketchgröße: 2.596 Bytes (von einem Maximum von 32.256 Bytes)







analog_in_GP2Y0A21YK0F | Arduino 1.0.3

Datei Bearbeiten Sketch Tools Hilfe



analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F to pin 0

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```

COM11

Senden

155
140
150
151
155
128
150
151
147
146
150
148
146
148
151

☒ Automatisch scrollen

Kein Zeilenende

115200 baud

Upload abgeschlossen

Binäre Sketchgröße: 2.596 Bytes (von einem Maximum von 32.256 Bytes)



Analoge Distanzsensordaten verarbeiten



```
analog_in_GP2Y0A21YK0F
```

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```

Funktionsweise

Fotowiderstand

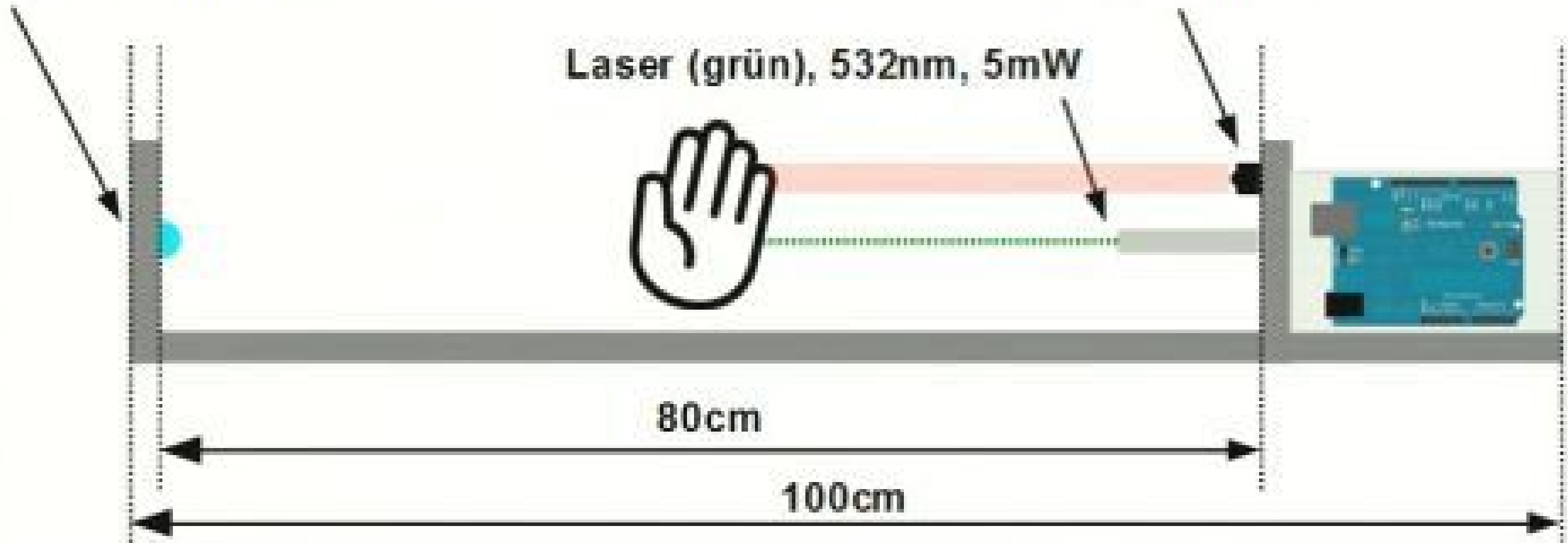
Distanzsensor

Laser (grün), 532nm, 5mW

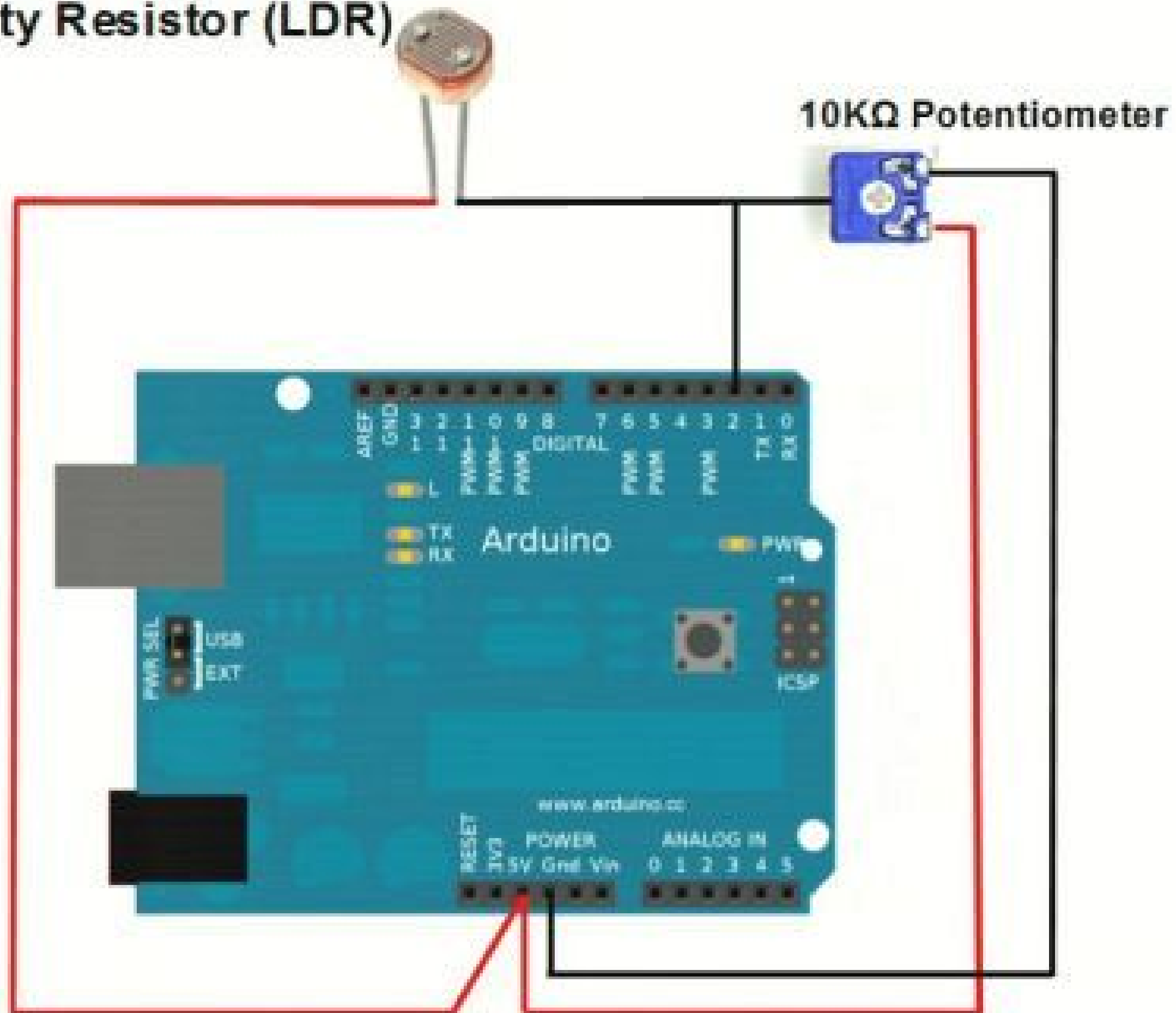


80cm

100cm



Light Density Resistor (LDR)



analog_in_GP2Y0A21YK0F

```
#define LOOPDELAY 250
#define PITCHINPUT 0 //Connect the GP2Y0A21YK0F sensor output to this analog pin

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) analogRead(PITCHINPUT));
  delay(LOOPDELAY);
}
```

Upload abgeschlossen

Binäre Sketchgröße: 2.596 Bytes (von einem Maximum von 32.256 Bytes)

1

Arduino Uno on COM11

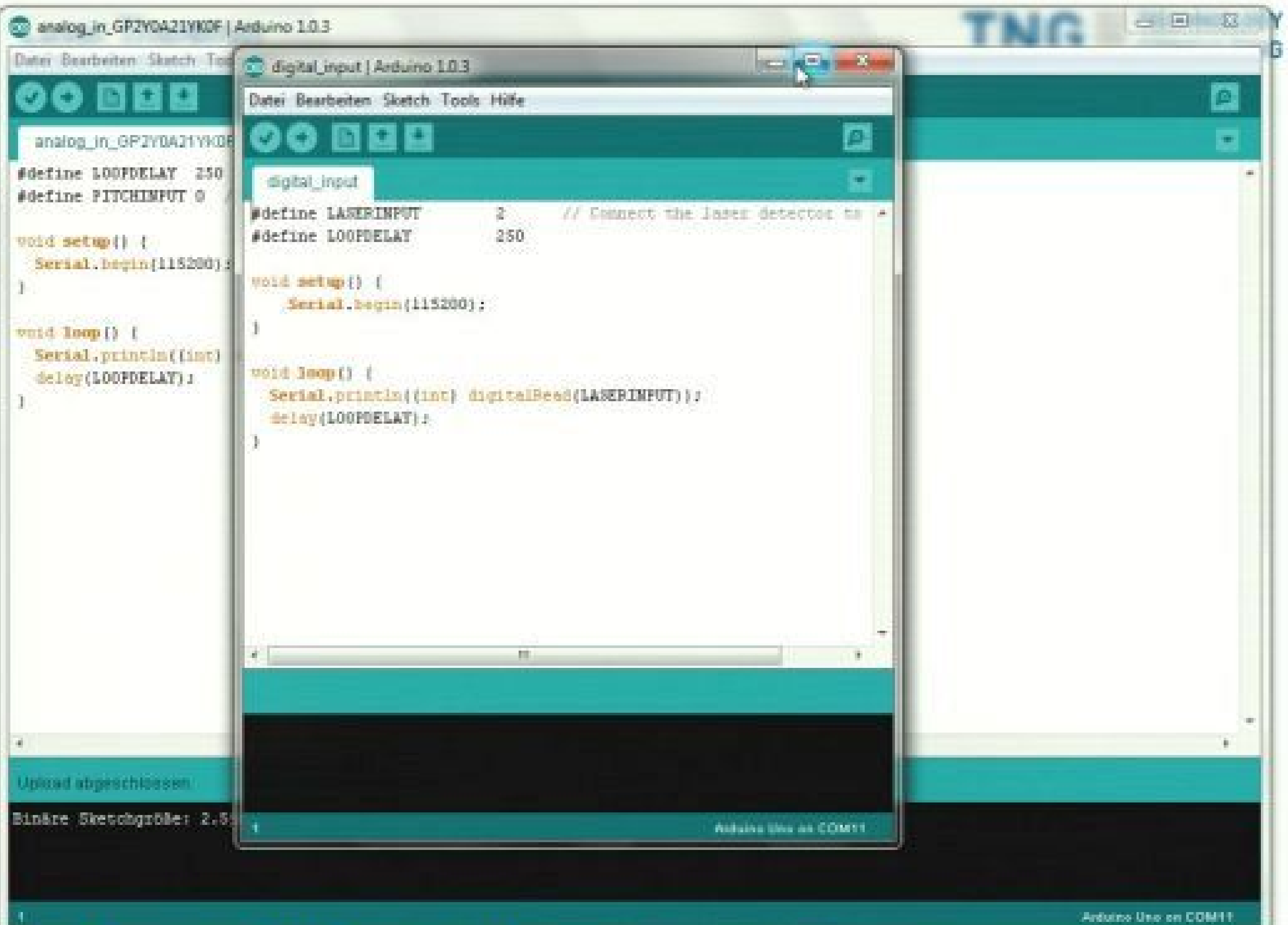
Neu	Strg+N
Öffnen...	Strg+O
Sketchbook	
Beispiele	
Schließen	Strg+W
Speichern	Strg+S
Speichern unter...	Strg+Umschalt+S
Upload	Strg+U
Upload mit Programmer	Strg+Umschalt+U
Papierformat	Strg+Umschalt+P
Drucken	Strg+P
Einstellungen	Strg+Comma
Beenden	Strg+Q

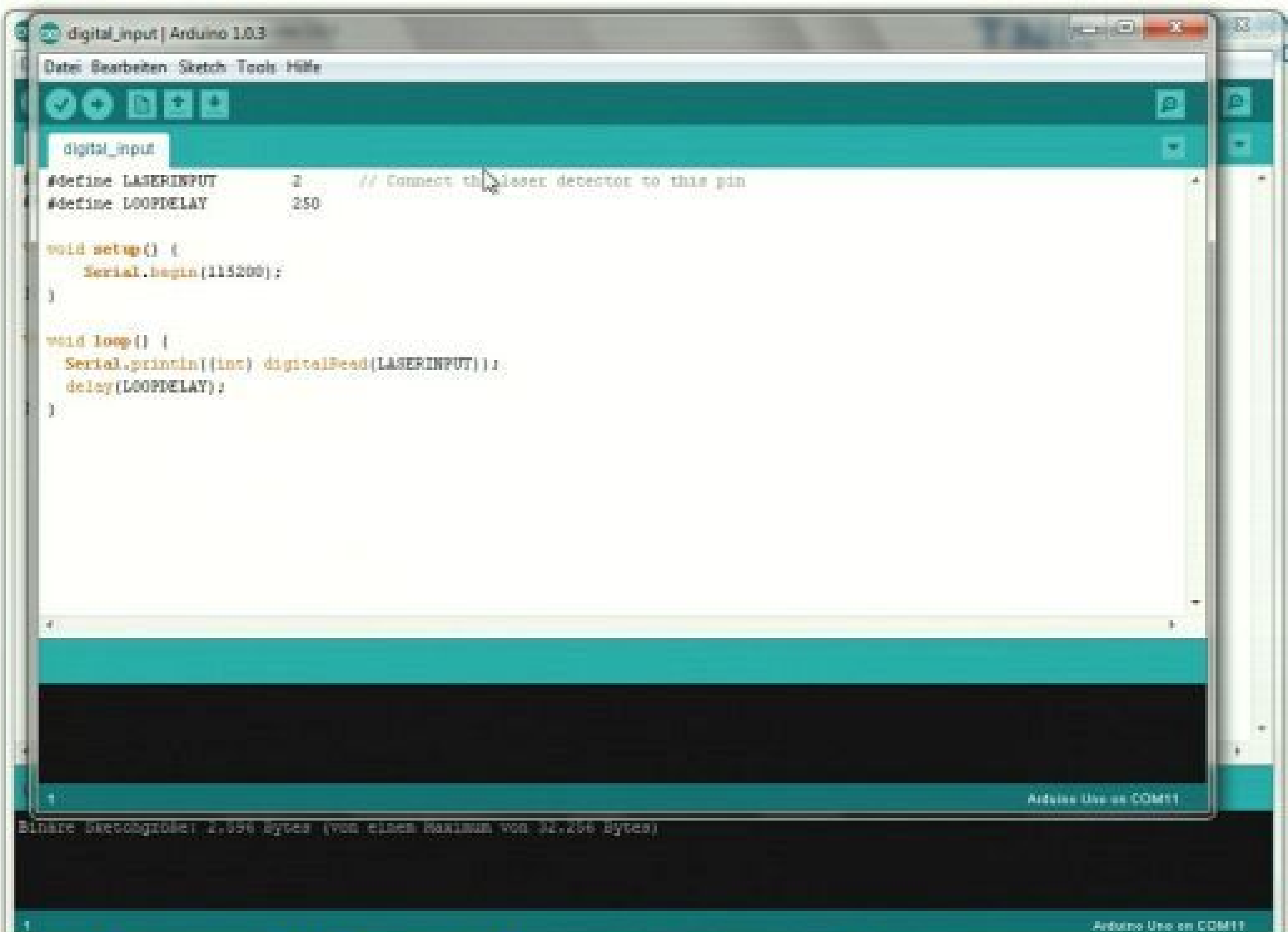
GP2Y0A21YK0F sensor output to this analog pin

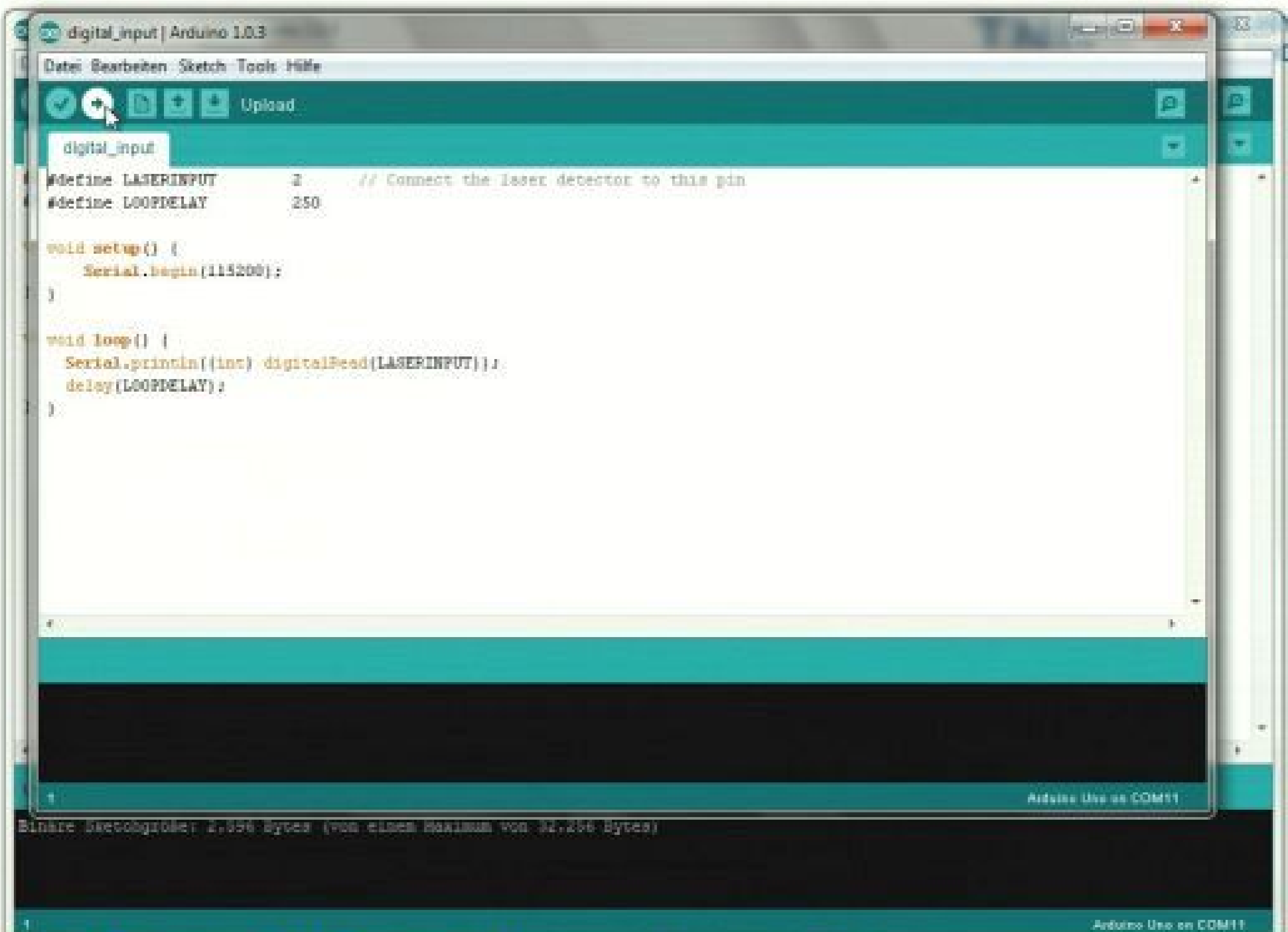
INPUT);

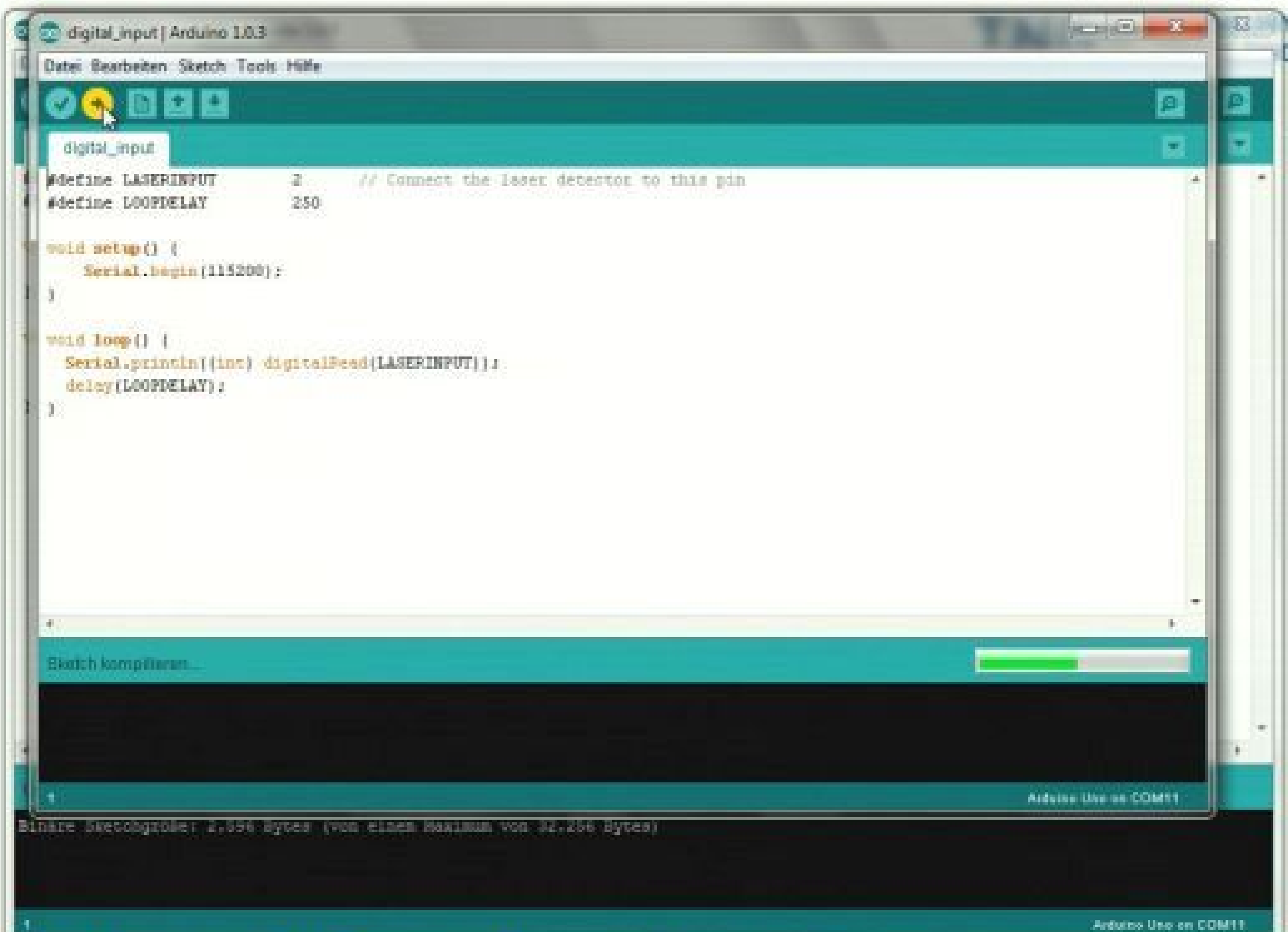
Upload abgeschlossen

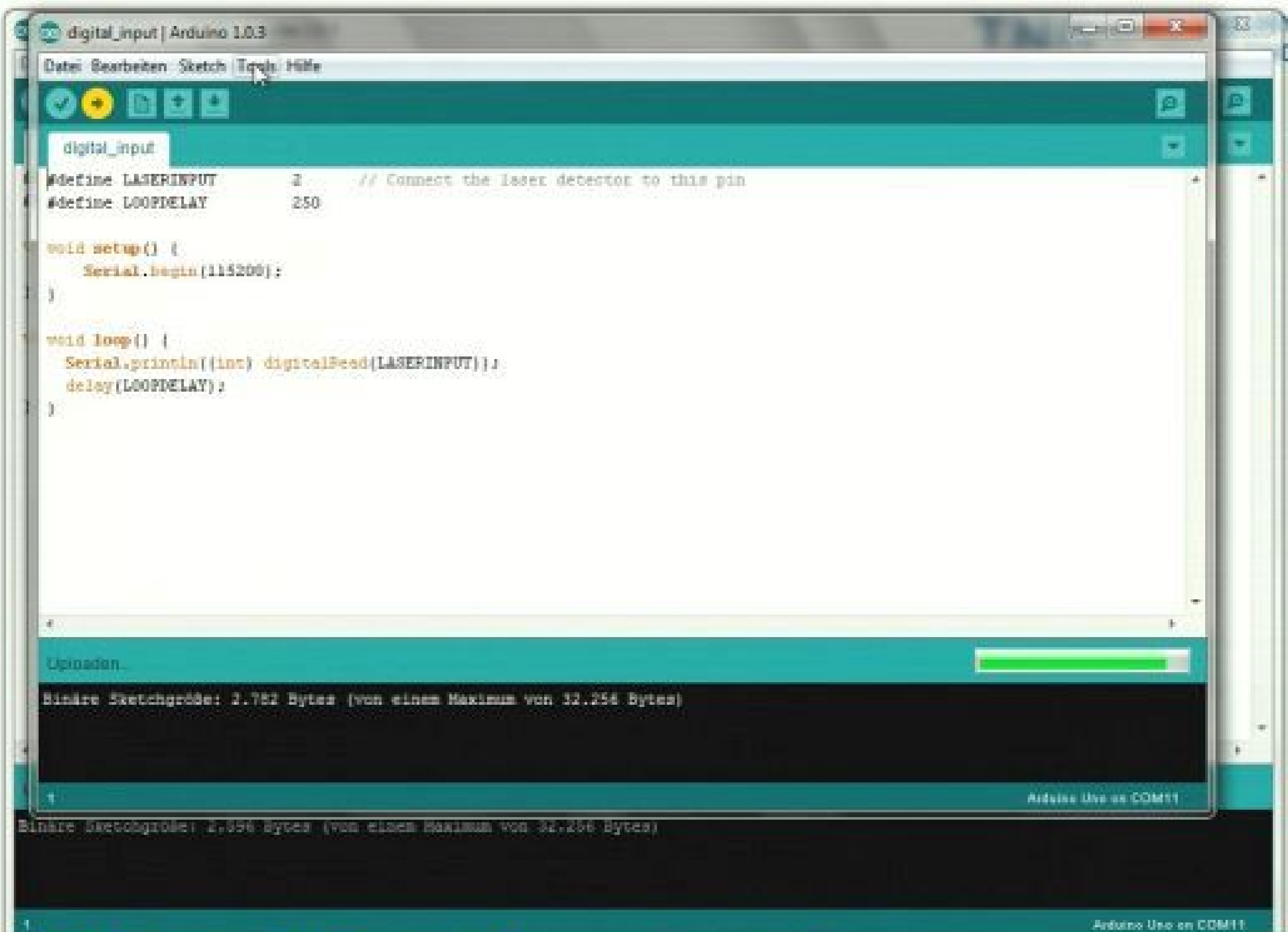
Binäre Sketchgröße: 2.596 Bytes (von einem Maximum von 32.256 Bytes)

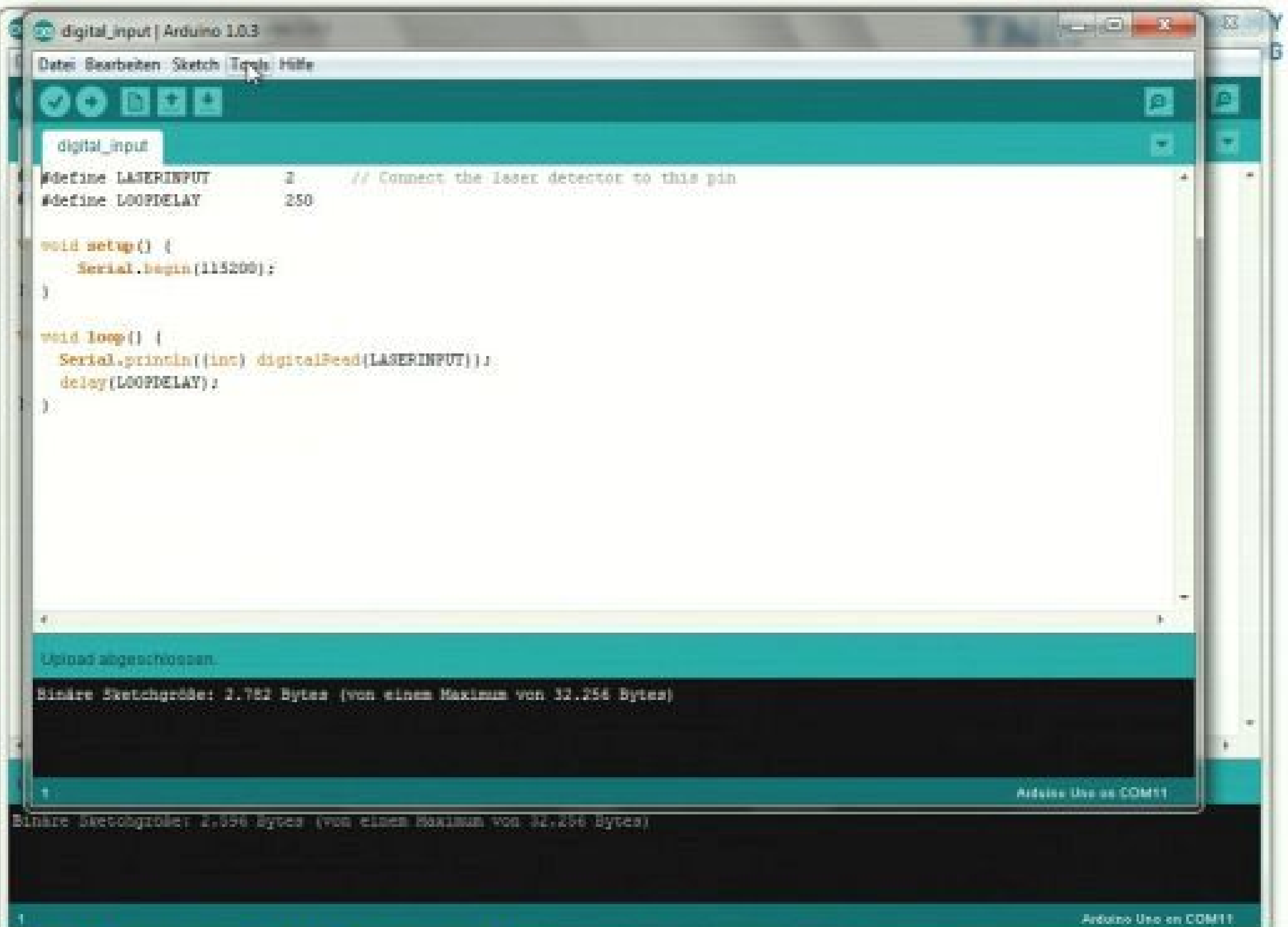


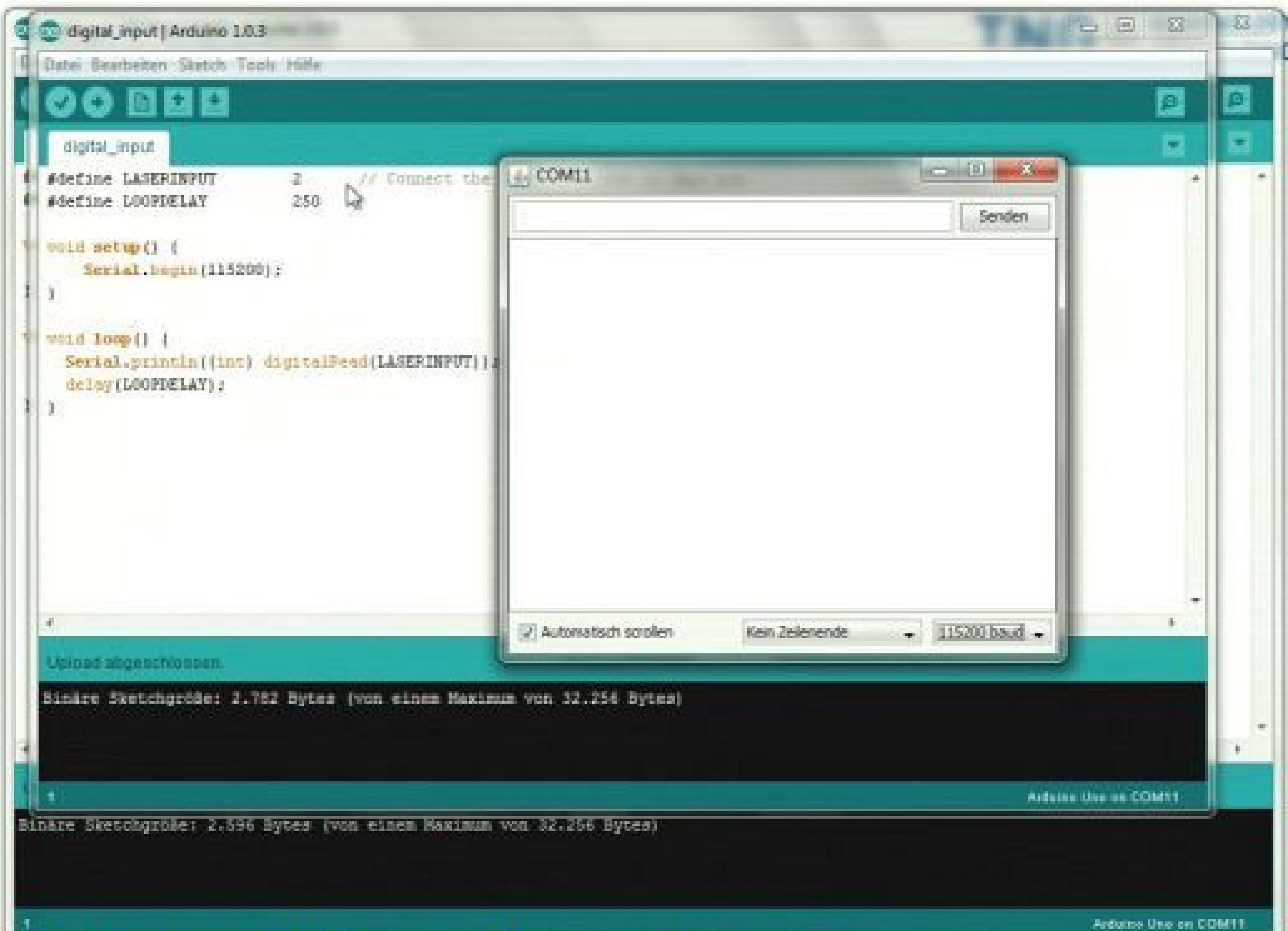


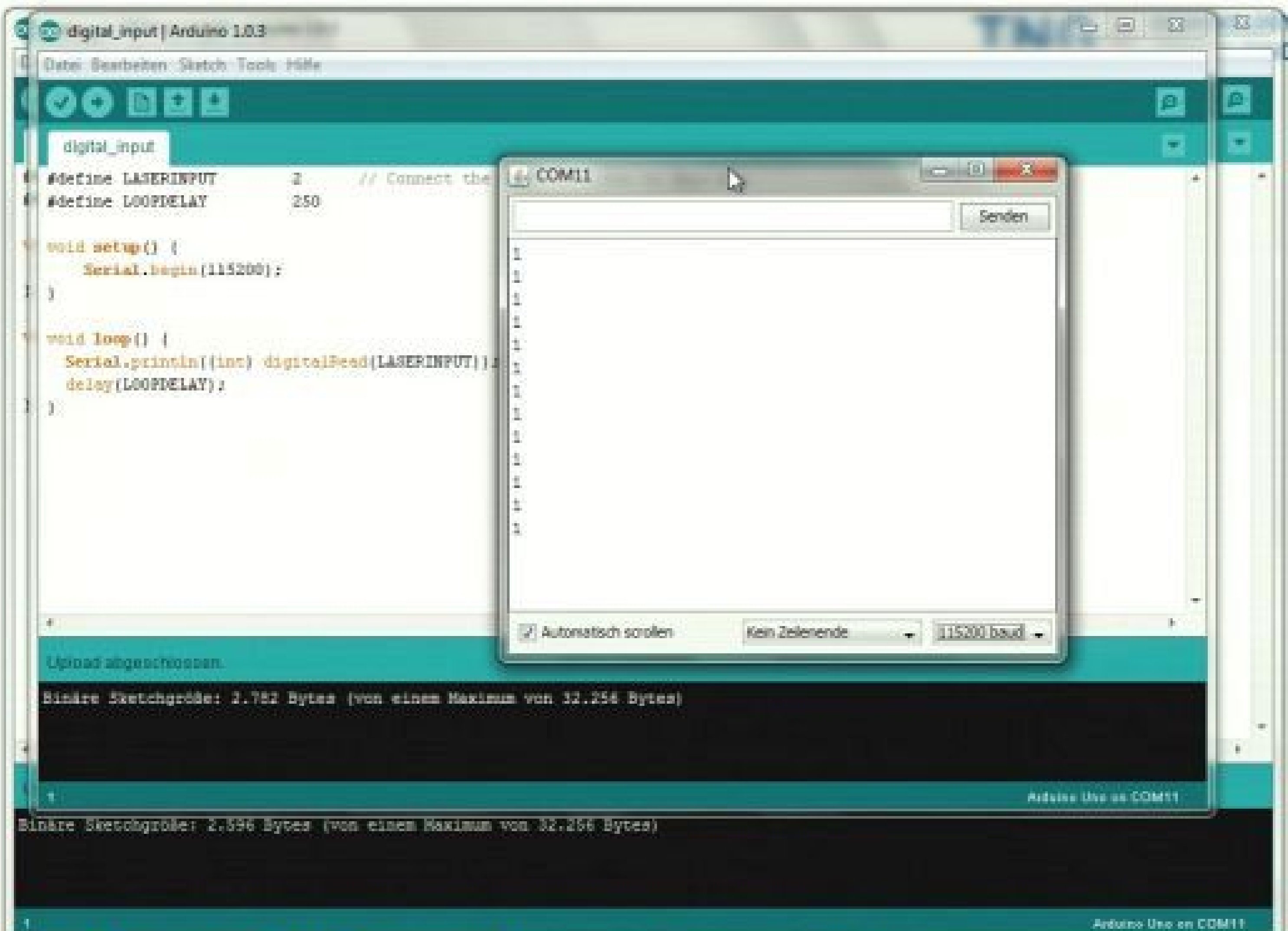


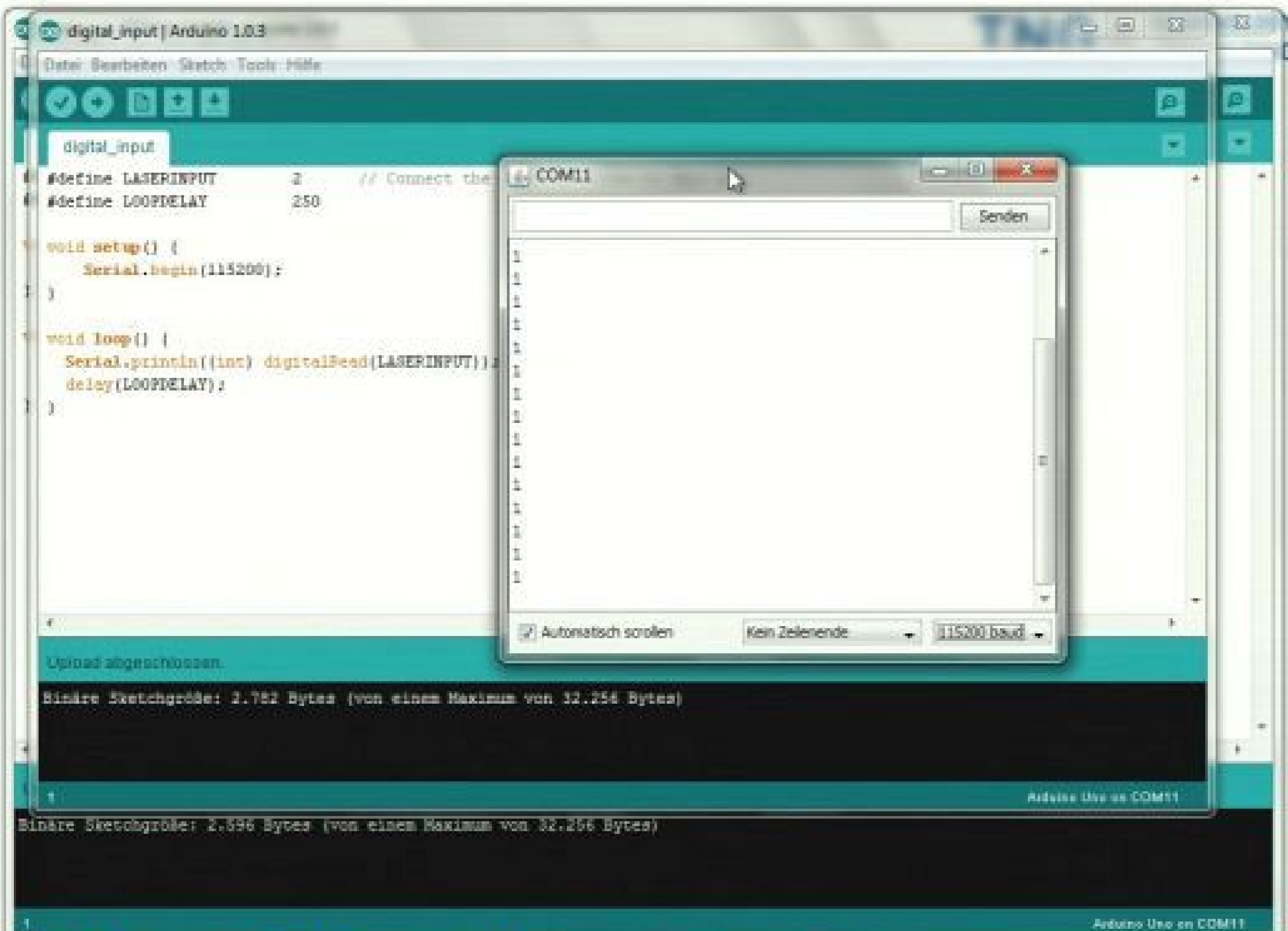


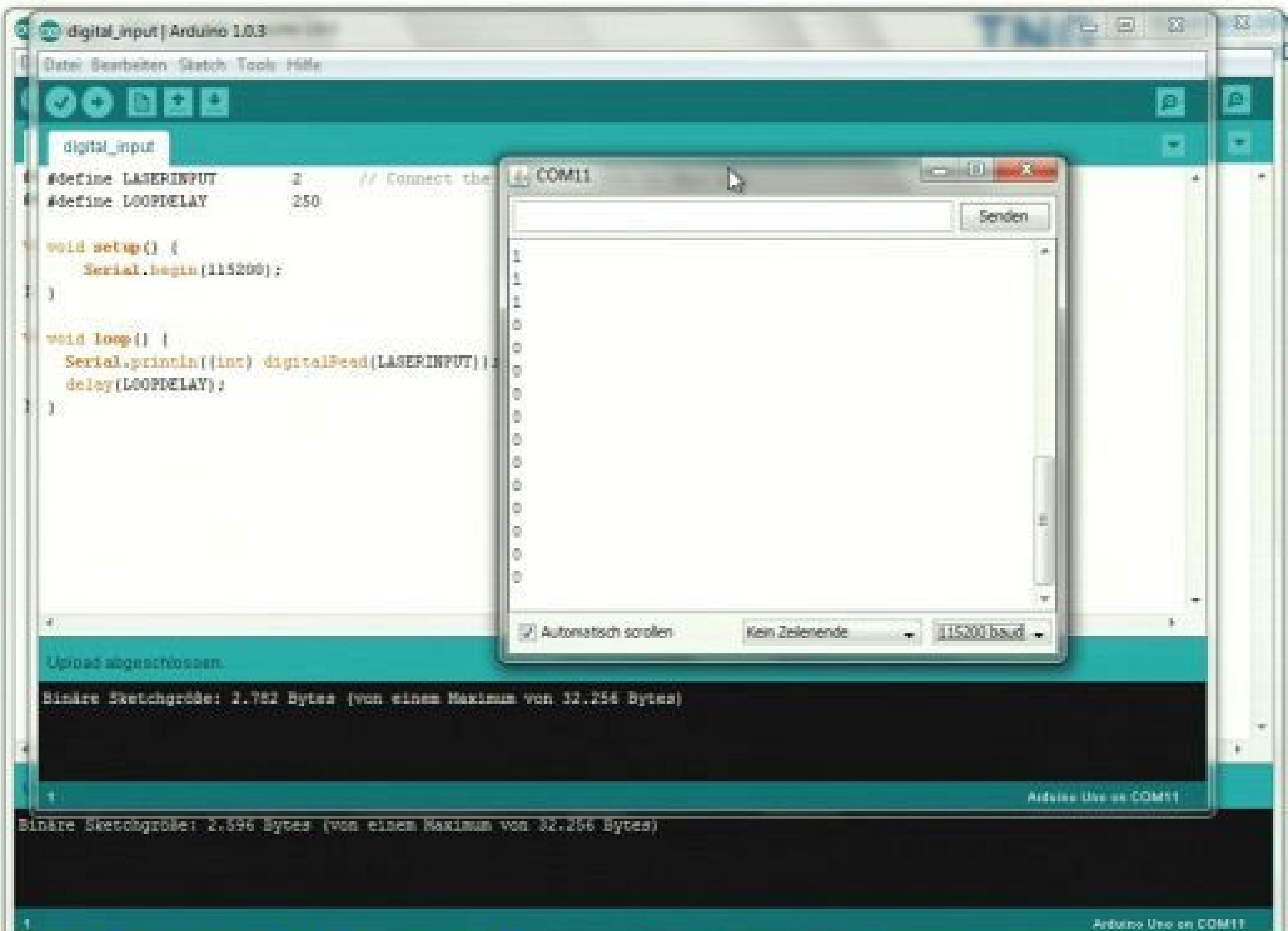


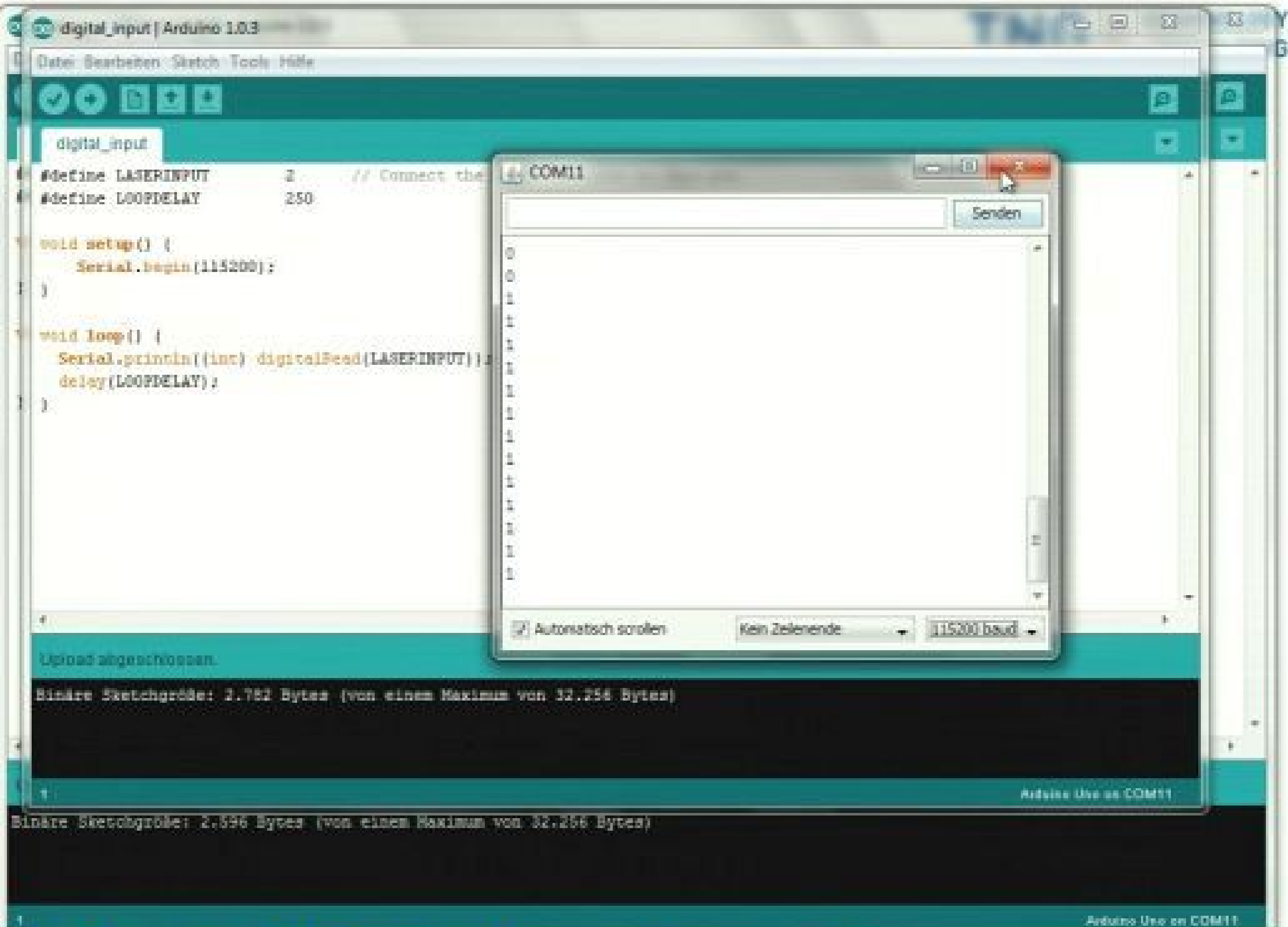


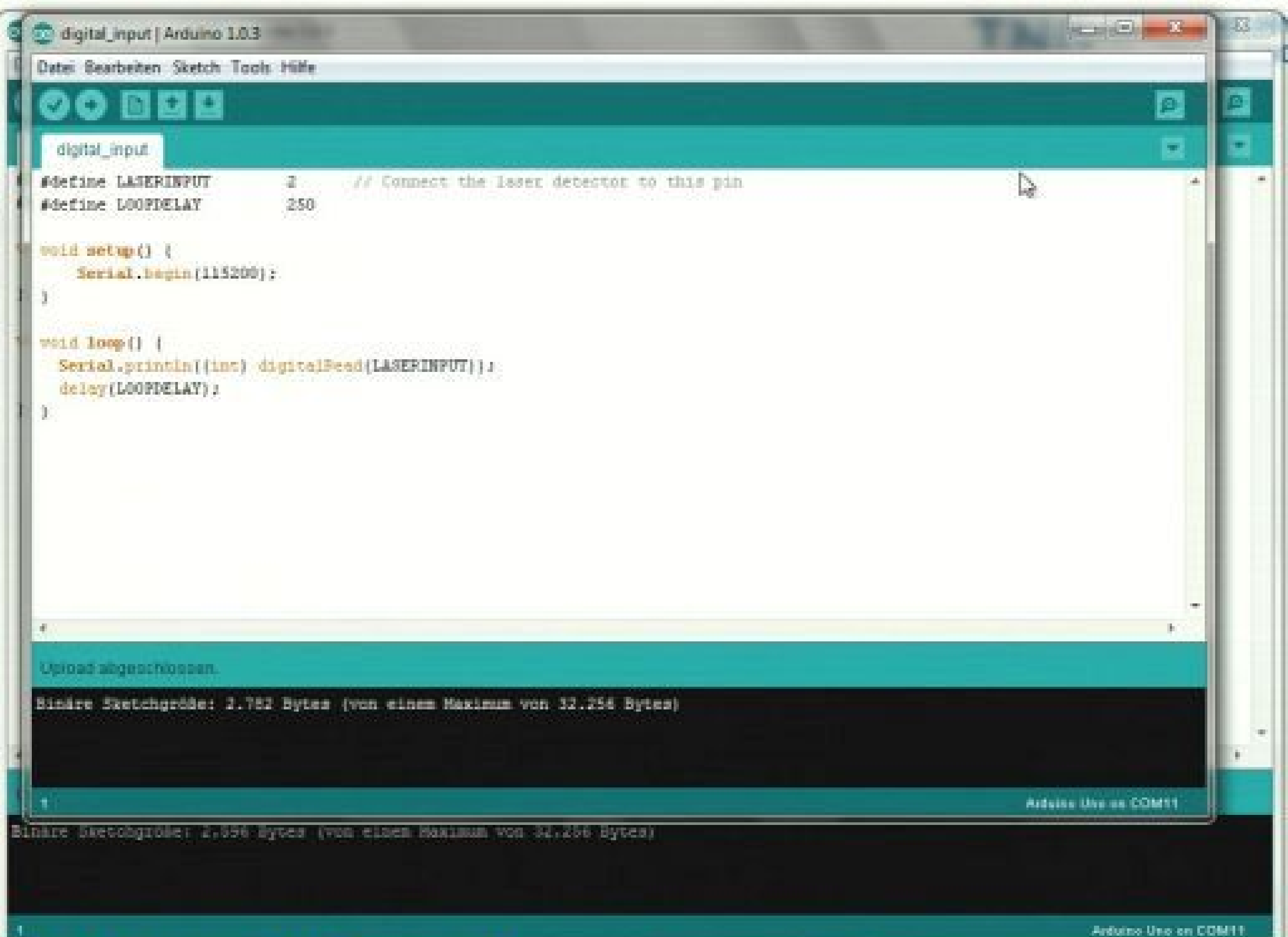












```
#define LASERINPUT 2 // Connect the laser detector to this pin
#define LOOPDELAY 250

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println((int) digitalRead(LASERINPUT));
  delay(LOOPDELAY);
}
```

Upload abgeschlossen.

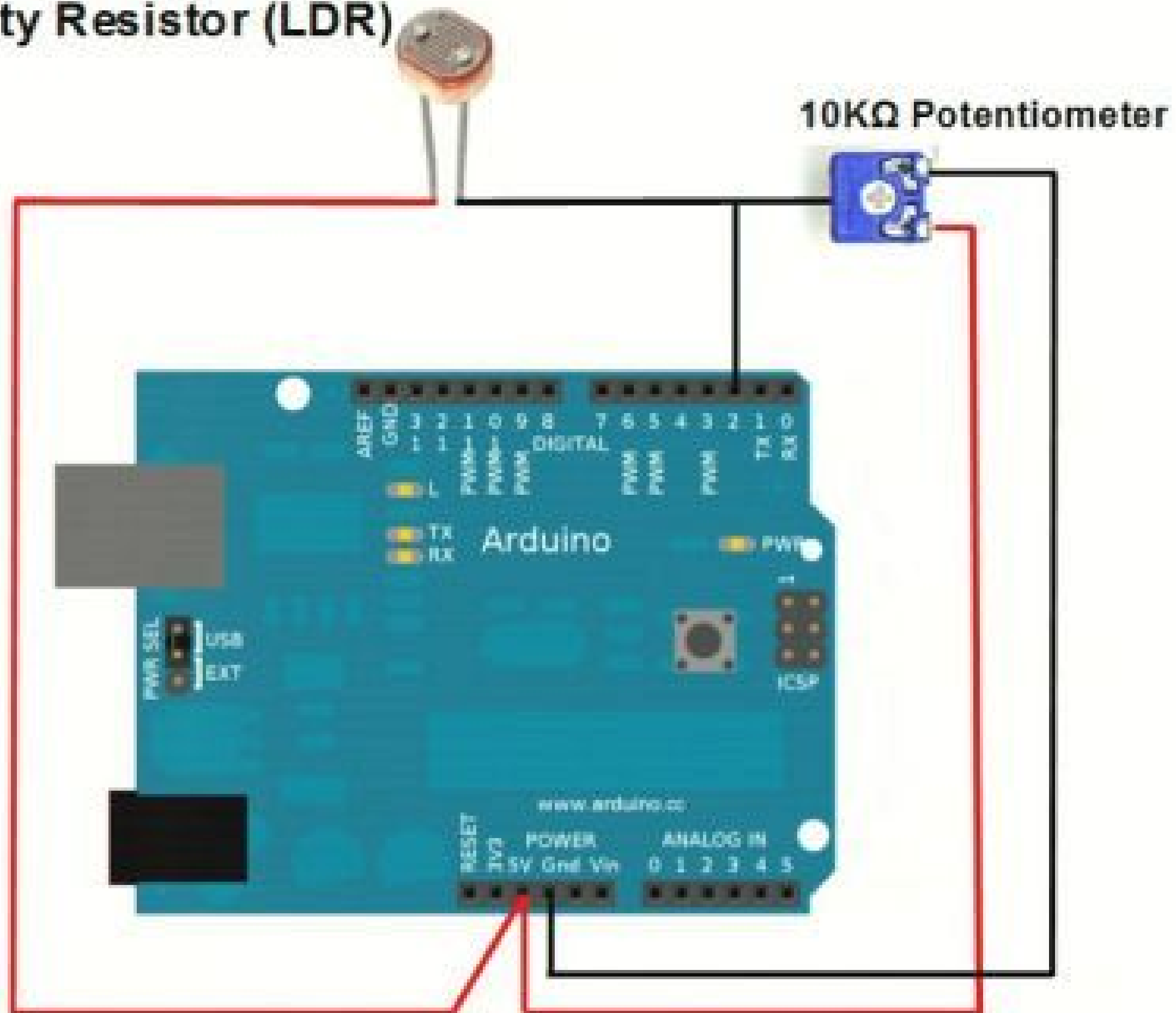
Binäre Sketchgröße: 2.782 Bytes (von einem Maximum von 32.256 Bytes)

Arduino Uno on COM11

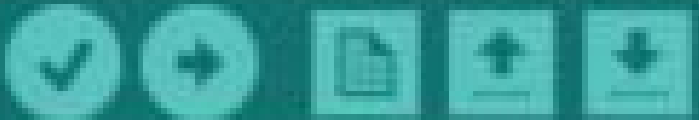
Binäre Sketchgröße: 2.856 Bytes (von einem Maximum von 32.256 Bytes)

Arduino Uno on COM11

Light Density Resistor (LDR)



Digitale Sensordaten verarbeiten



```
digital_input

#define LASERINPUT      2      // Connect the laser detector to this pin
#define LOOPDELAY        250

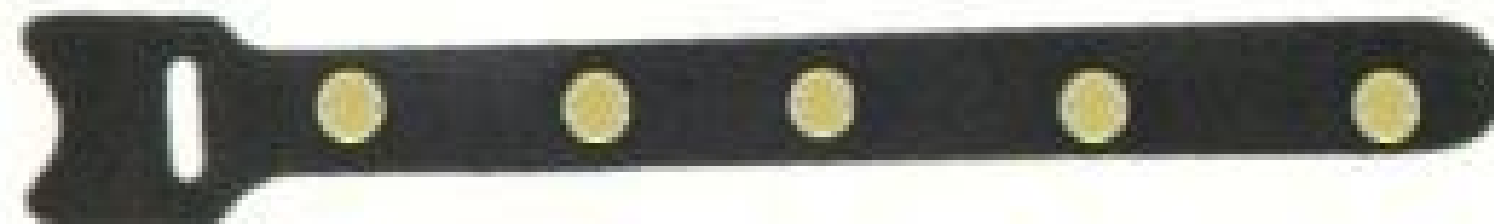
void setup() {
    Serial.begin(115200);
}

void loop() {
    Serial.println((int) digitalRead(LASERINPUT));
    delay(LOOPDELAY);
}
```

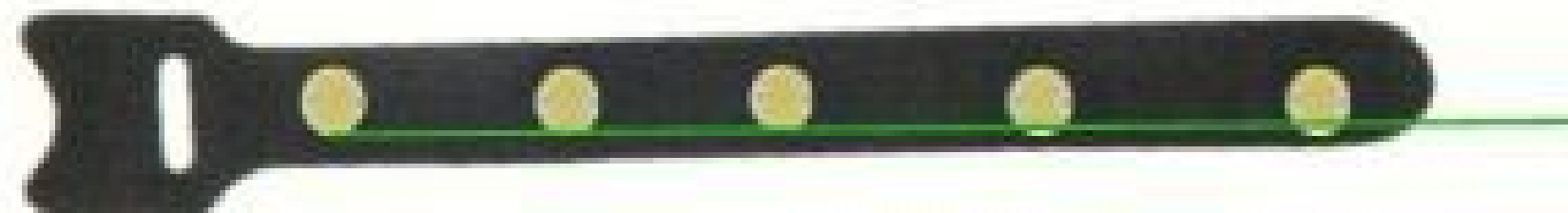
Agenda

- Einführung in die Arduino-Plattform
- Was ist das – MIDI?
- Beat Sneakers
- Laser Theremin
- **Lavalampe**
- Der menschliche Körper als Controller LEAP

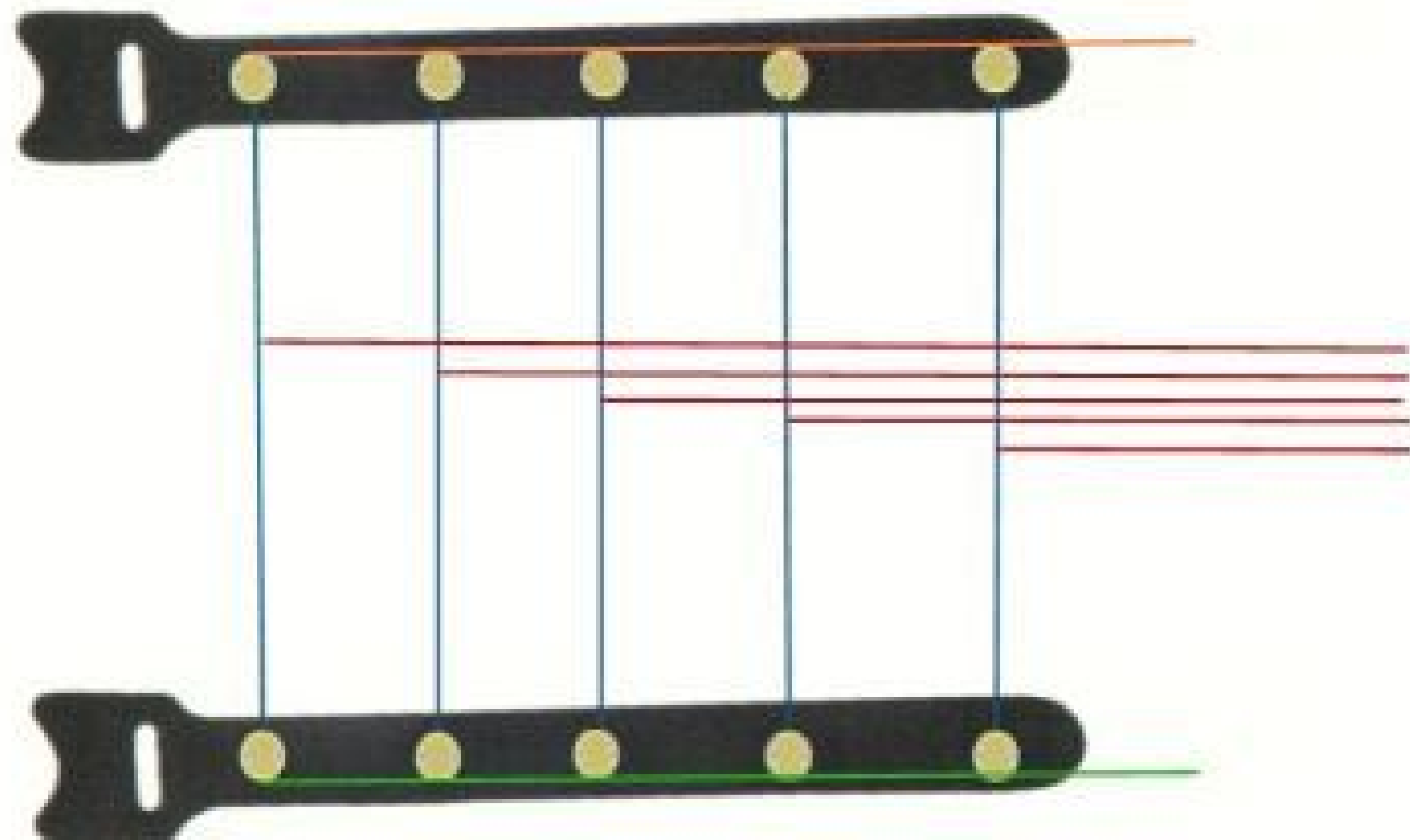
Klettband und LDRs miteinander verbinden



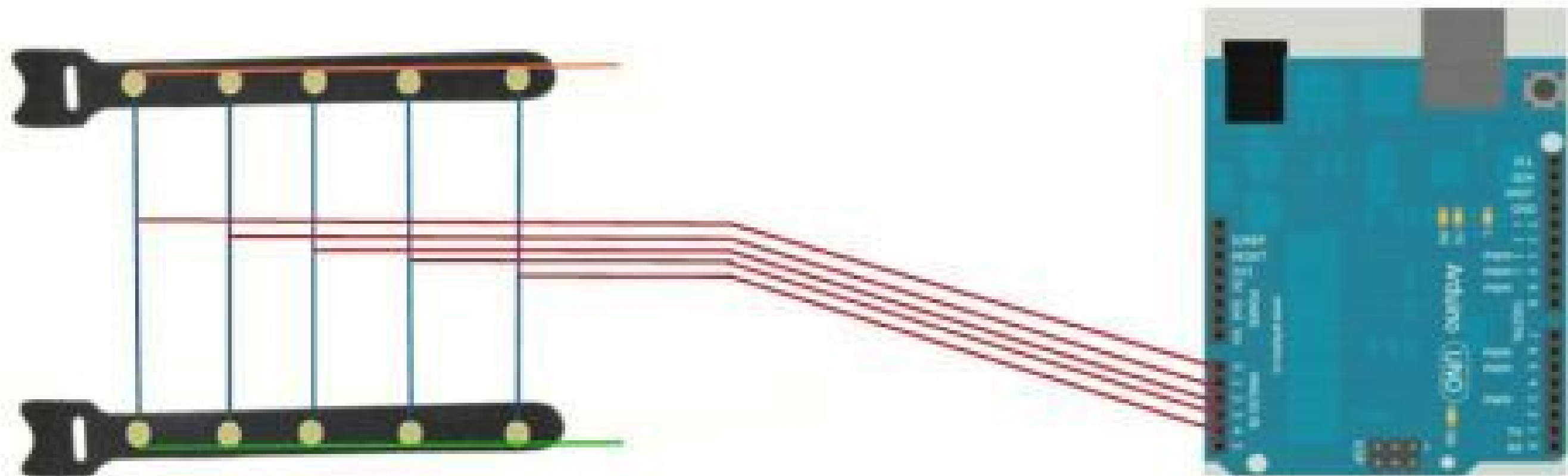
Nun folgt die Energieversorgung



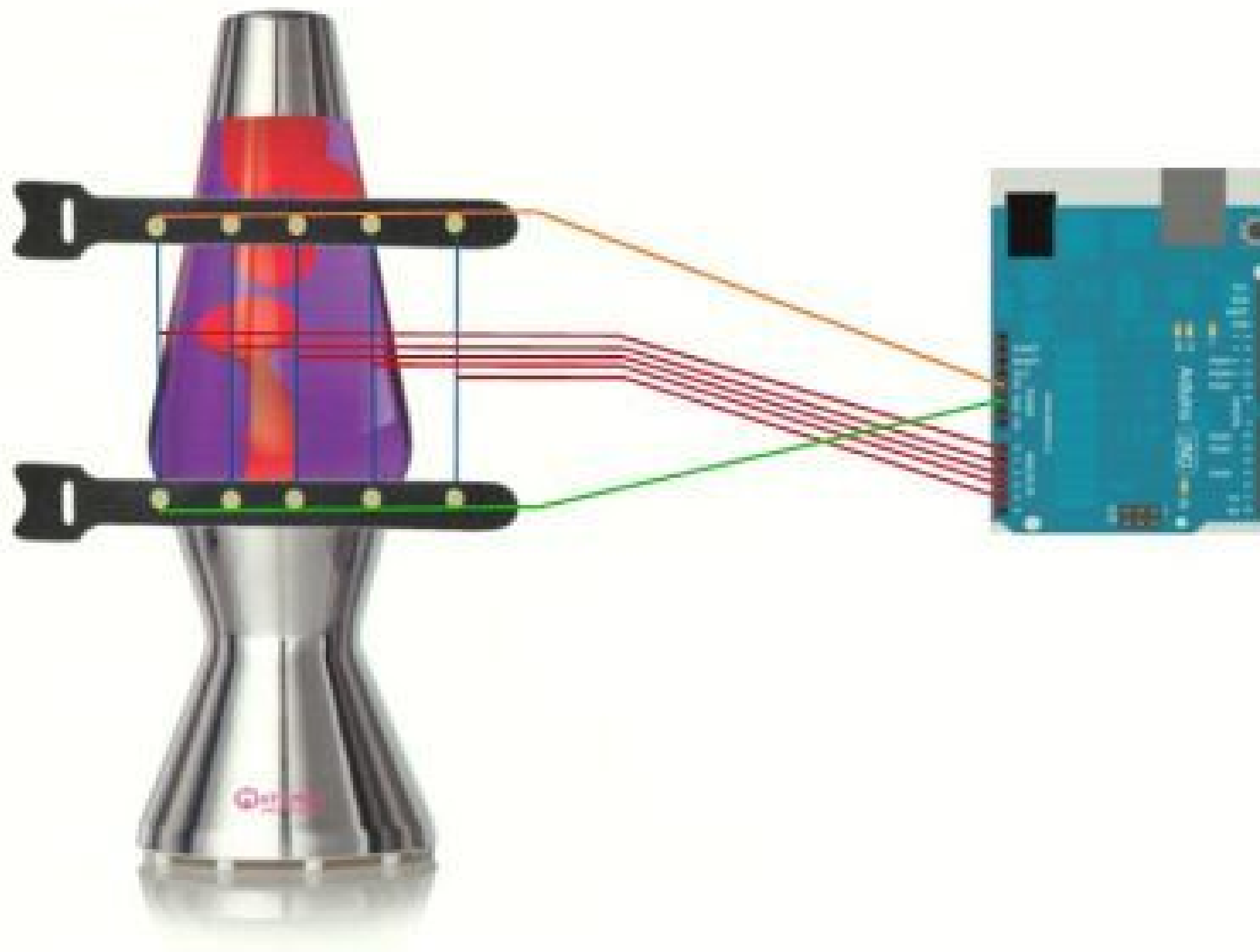
Leitungen für die Wertemessung anschliessen



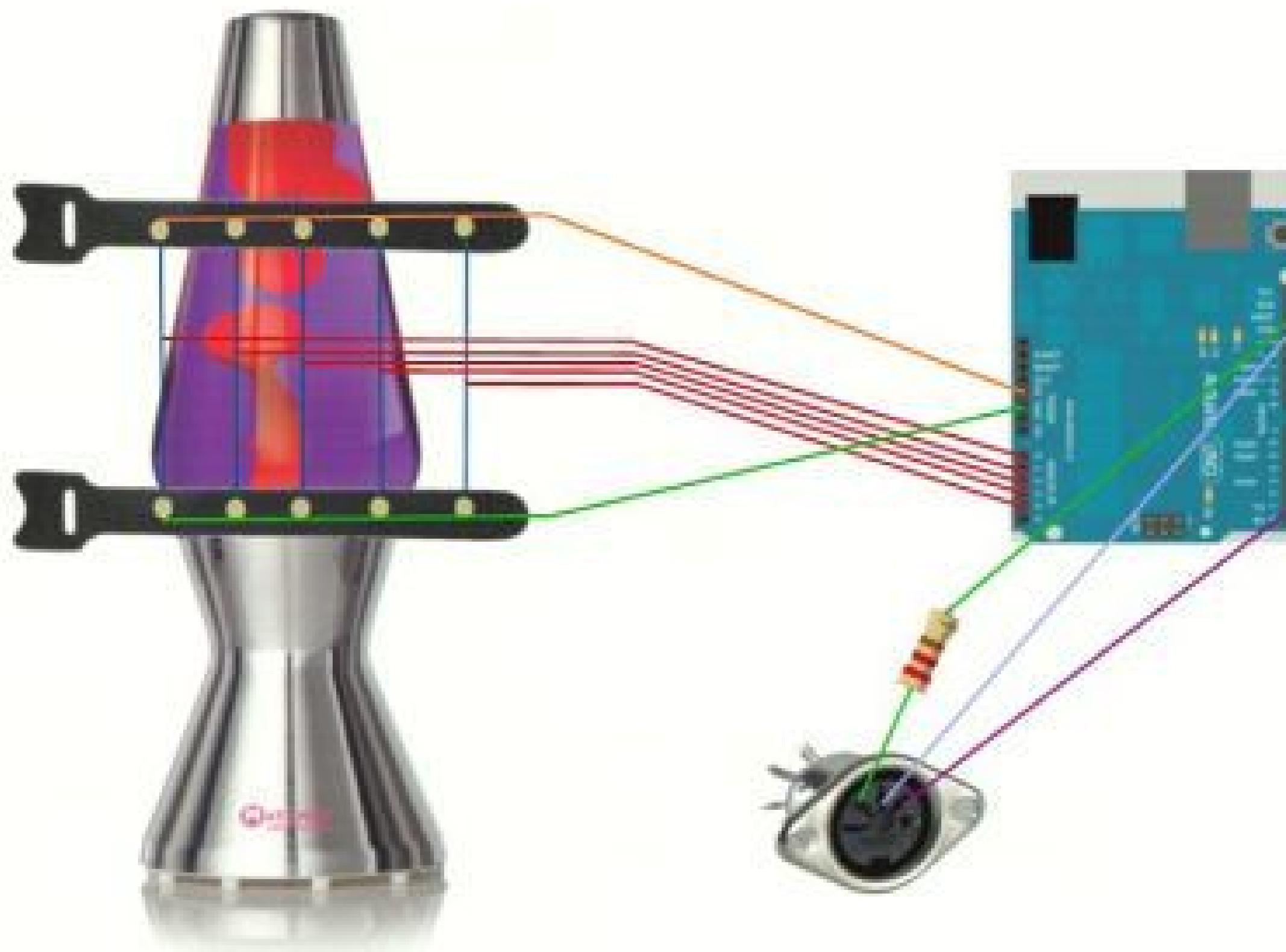
Die Messleitungen verbinden



Lavalampe bestücken



Fast fertig!



```
// run the monitor procedure for each controller in turn
for(int i=0;i<6;++i)
  C[i].Monitor(startupFlag);

delay(1);
```

```
void loop() { ... }
```

Den jeweiligen
Sensor auslesen

```
// read the raw analog value 0-1023
int value = analogRead(analogPin);

// add the value into the history buffer
long tot = 0;
for(int j=0; j<BUflen-1;++j)
{
  history[j] = history[j]+1;
  tot += history[j];
}
history[BUflen-1] = value;
tot+=value;

// if the history buffer is not yet
// full then return
if(!startup)
  return;

// get the average value from history
value = ((float)tot/BUflen);

// remember highest and lowest values
if((minReading == -1) || (minReading > value))
  minReading = value;
if((maxReading == -1) || (maxReading < value))
  maxReading = value;

// get the range of known readings
int range = maxReading - minReading;
if(range < 1)
  range = 1;

// scale the current value into the range
int newValue = 127.0 * (((float)value - (float)minReading) / (float)range);
```

Das 3-Byte
MIDI-Signal

```
// has the value changed?
if(newValue != currentValue)
{
  // light the LED
  digitalWrite(13,HIGH);

  // write out the CC message
  Serial.write(0xB0 | midiChannel);
  Serial.write(midiController);
  Serial.write(newValue);

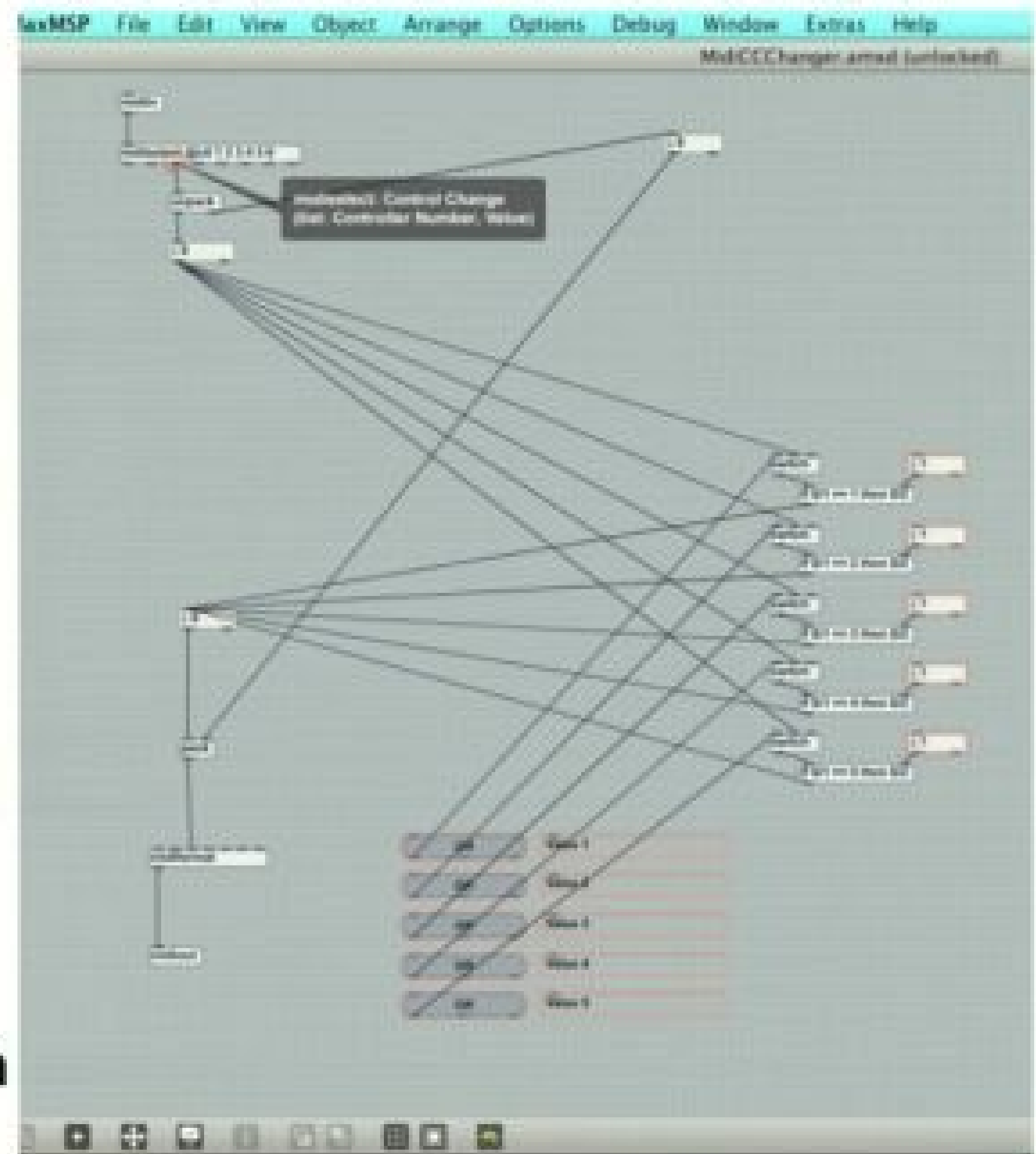
  // unlight the light
  digitalWrite(13,LOW);

  // remember the value
  currentValue= newValue;
}
```

Steuerung



MIDI CC-Wert

An-/Ausschalten
des Sensors

Agenda

- Einführung in die Arduino-Plattform
- Was ist das – MIDI?
- Beat Sneakers
- Laser Theremin
- Lavalampe

- **Der menschliche Körper als Controller**

LEAP

Es war einmal ...



<http://img.sagepub.com/public/images/1/10/nintendo-wii-remote-controller-01.jpg>

Nintendo WiiMote



<http://img.sagepub.com/public/images/1/10/microsoft-kinect-01.jpg>

Microsoft Kinect

Da kommt was auf uns zu

CREATIVE



Creative Interactive Gesture Camera



Intel Perceptual Programming SDK

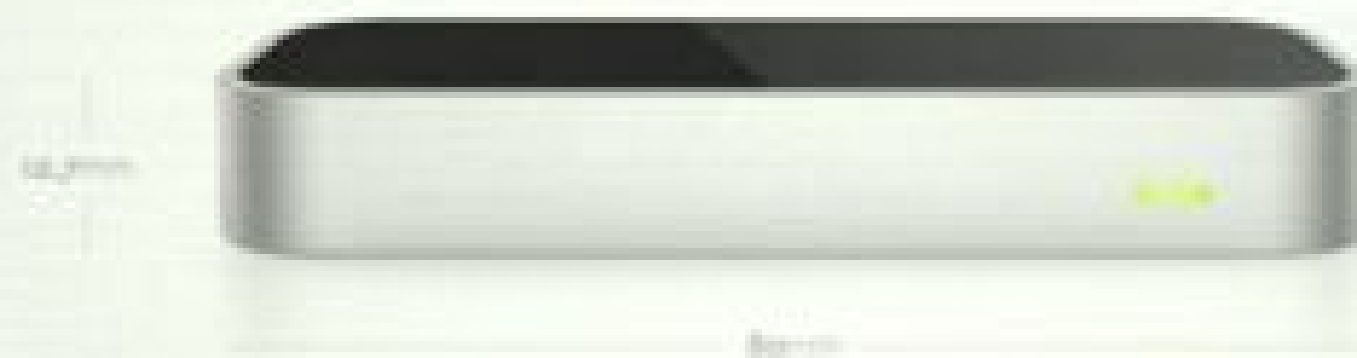
Das war noch nicht alles

<http://indiepenario.it/sites/default/files/styles/wide/public/field/image/leap-motion.png?itok=GS4HQS5T>

LEAP



Leap Motion



<https://leapmotion.com/>

Leap Motion

- USB-Controller mit zwei Kameras und drei Infrarot-LEDs
 - 3D-Tiefentracking (hemispherisch) mit einer Distanz bis zu einem Meter
- Tracken von Fingern (oder fingerähnlichen Gegenständen)
 - räumliche Genauigkeit von 0,01 mm
- USB-Verbindung wird benötigt
 - Wireless Version für zukünftige Modelle in Planung

Ein Blick in die Leap Motion

<http://www.creative-applications.net/wp-content/uploads/2012/11/leap2.jpg>



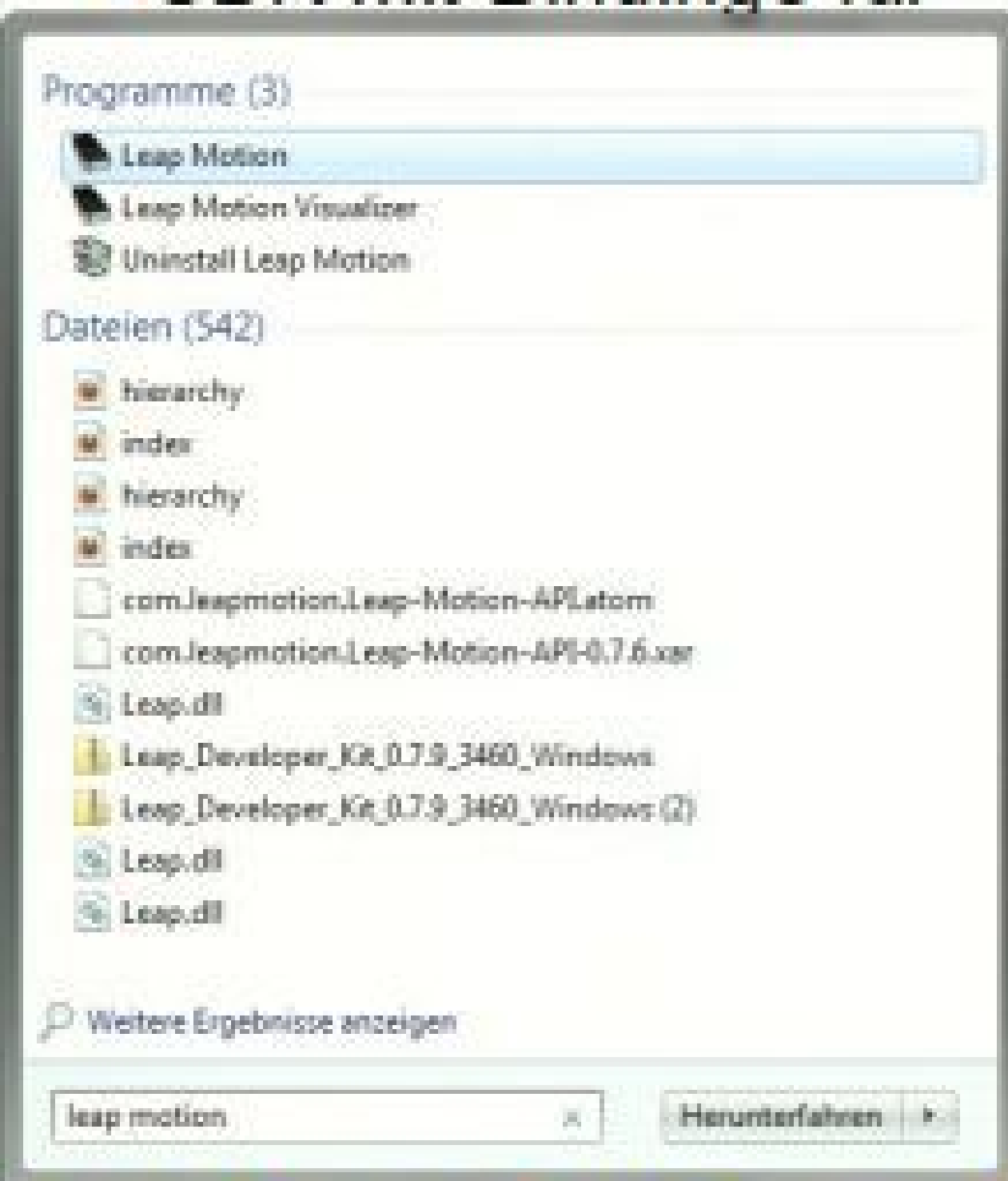
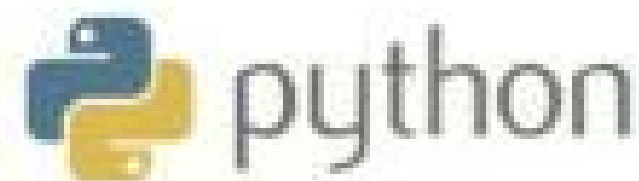
Leap Motion

- Bibliotheken für Windows, Linux, MacOS
- SDK mit Bindings für
 - Java
 - C++
 - C#
 - Obj-C
 - Python
 - JavaScript



Leap Motion

- Bibliotheken für Windows, Linux, MacOS
- SDK mit Bindings für



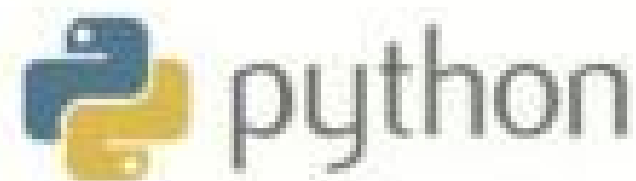
Leap Motion

- Bibliotheken für Windows, Linux, MacOS
- SDK mit Bindings für
 - Java
 - C++
 - C#
 - Obj-C
 - Python
 - JavaScript



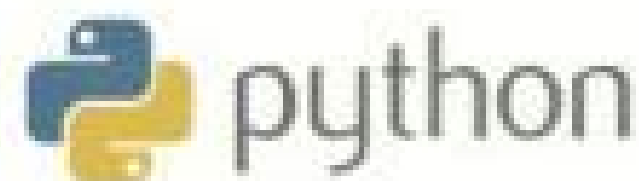
Leap Motion

- Bibliotheken für Windows, Linux, MacOS
- SDK mit Bindings für
 - Java
 - C++
 - C#
 - Obj-C
 - Python
 - JavaScript



Leap Motion

- Bibliotheken für Windows, Linux, MacOS
- SDK mit Bindings für
 - Java
 - C++
 - C#
 - Obj-C
 - Python
 - JavaScript



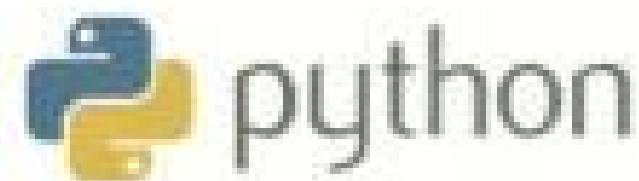
Bright Lighting Detected
The device has entered 'Robust Mode' to compensate for lighting conditions. A small diagnostics file has been saved to your user data directory.

Leap Motion Controller in robust mode



Leap Motion

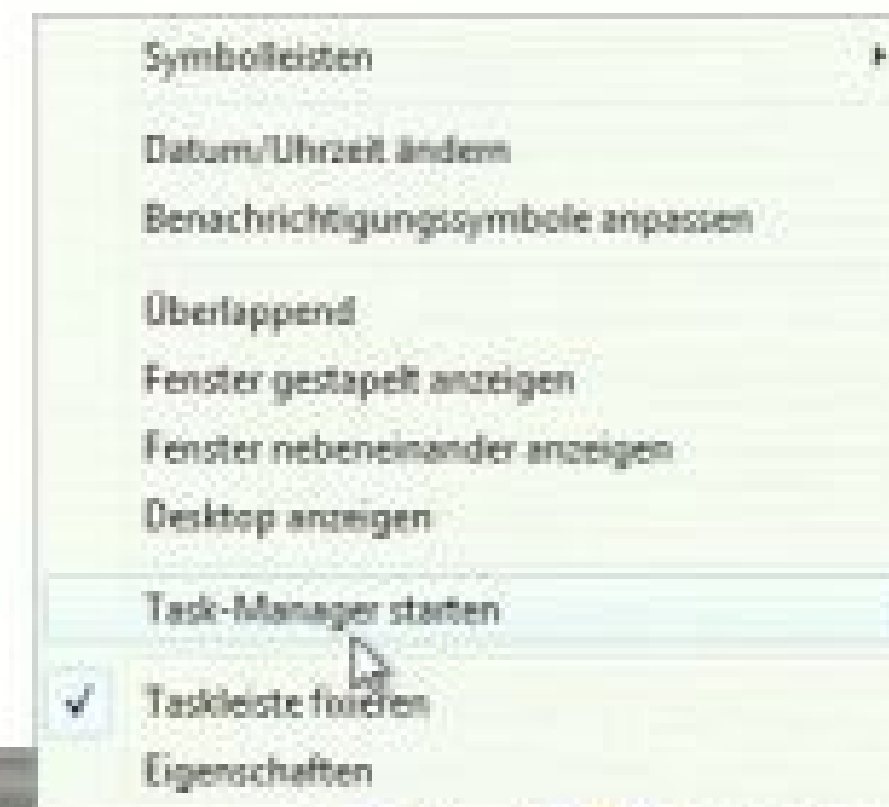
- Bibliotheken für Windows, Linux, MacOS
- SDK mit Bindings für
 - Java
 - C++
 - C#
 - Obj-C
 - Python
 - JavaScript



Bright Lighting Detected
The device has entered 'Robust Mode' to compensate for lighting conditions. A small diagnostics file has been saved to your user data directory.

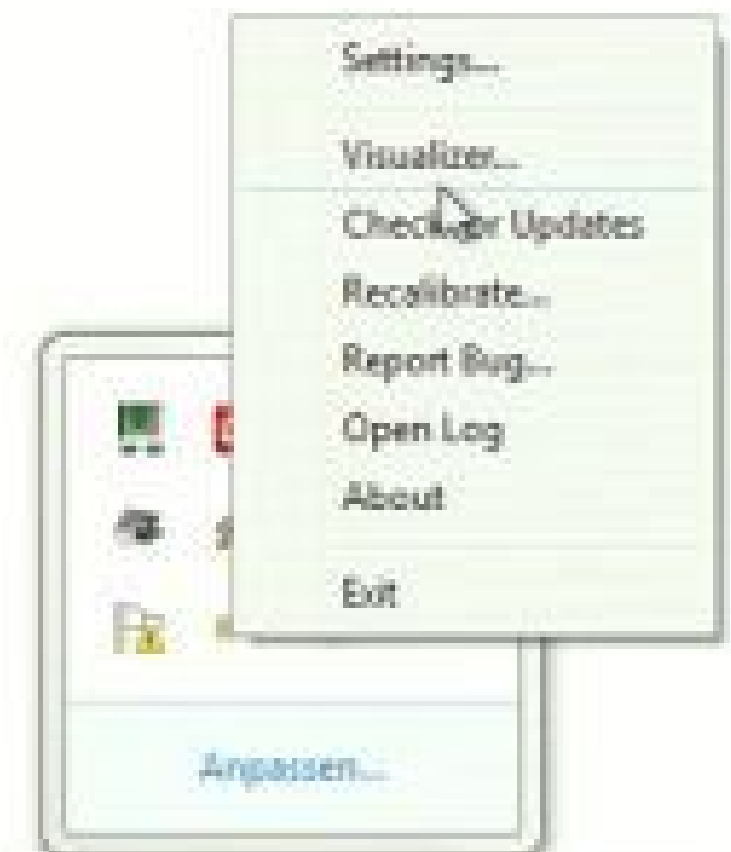
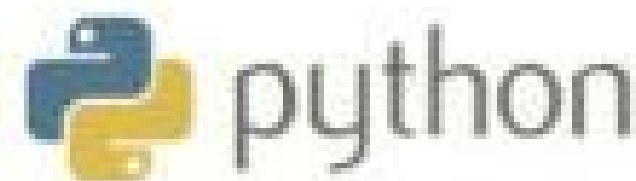
Leap Motion

- Bibliotheken für Windows, Linux, MacOS
- SDK mit Bindings für
 - Java
 - C++
 - C#
 - Obj-C
 - Python
 - JavaScript



Leap Motion

- Bibliotheken für Windows, Linux, MacOS
- SDK mit Bindings für
 - Java
 - C++
 - C#
 - Obj-C
 - Python
 - JavaScript



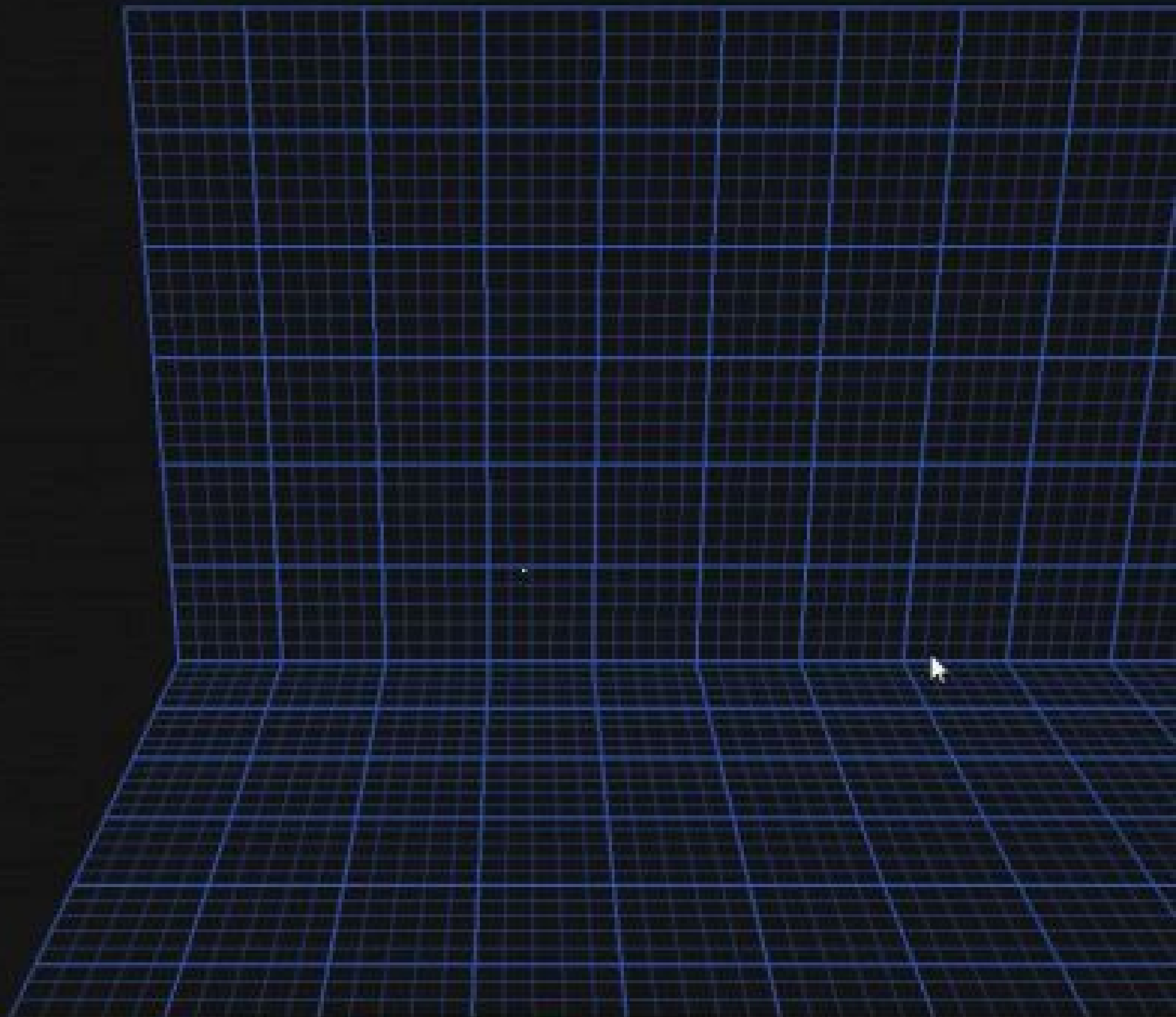
Der r

Leap Motion Visualizer

Leap

■ Bit

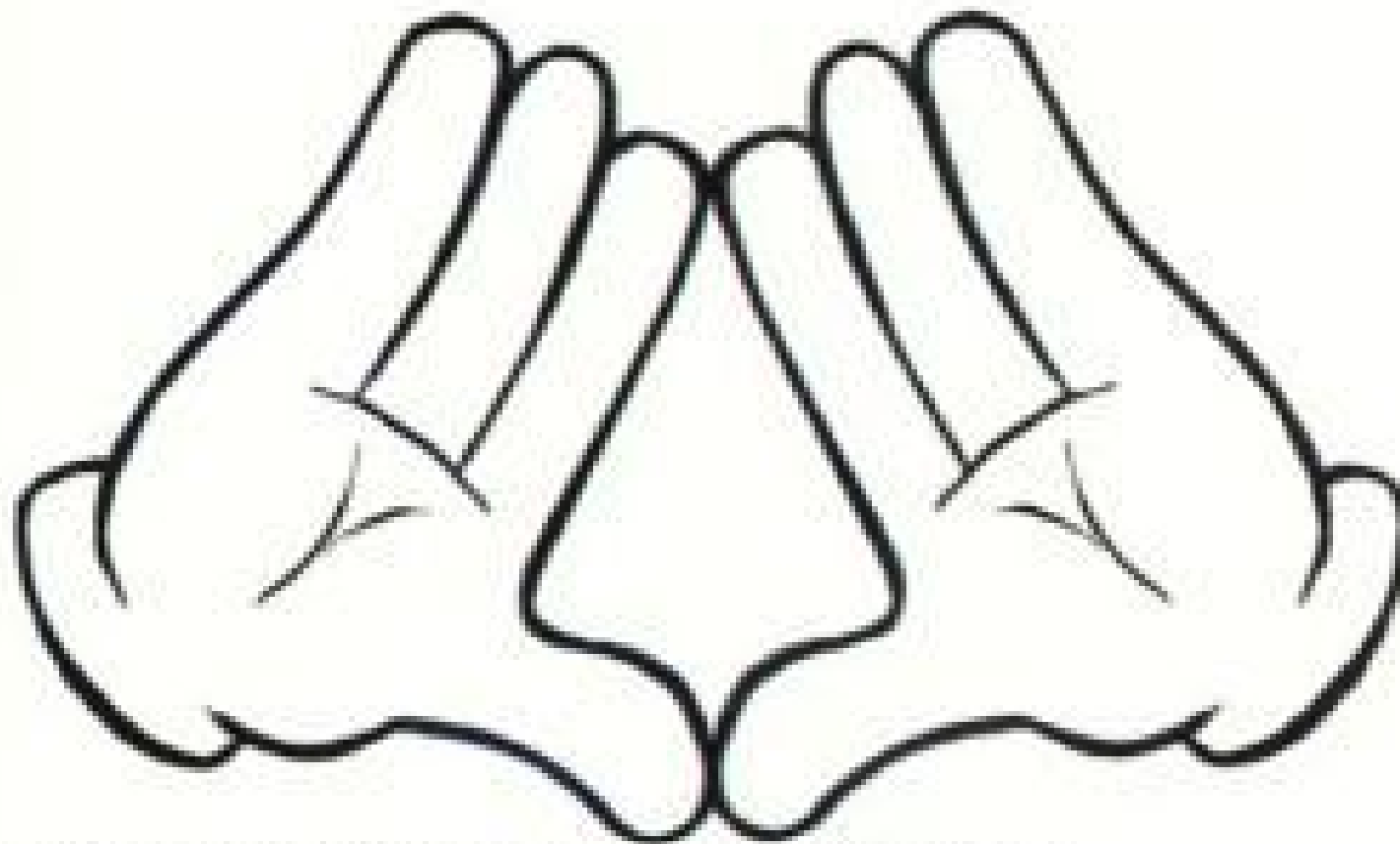
■ SD



Unsere Erfahrungen

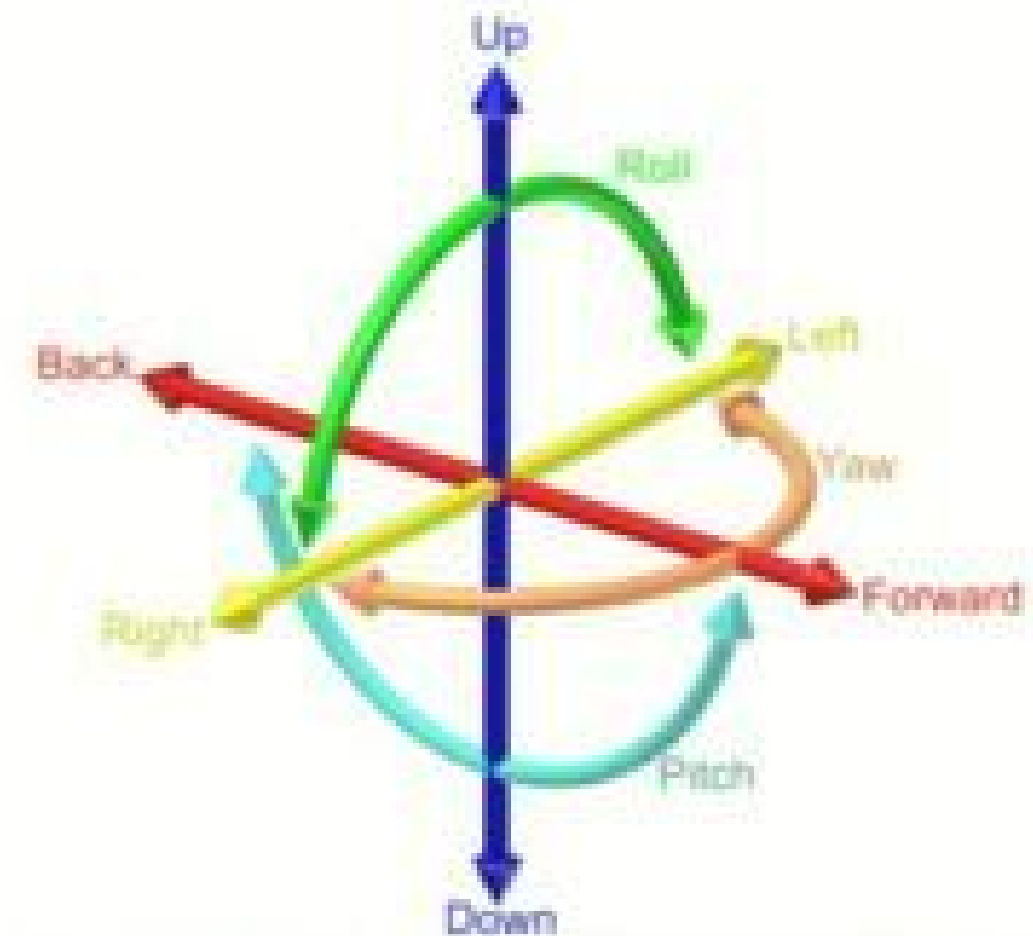
- Implementierung einer „Parrot AR.Drone“-Steuerung in Java zu zweit innerhalb von fünf Stunden
- Lokaler Socketserver vereinfacht Zugriff auf Trackingdaten
- für Webanwendungen geeignet

Welche Daten können erfasst werden



http://25-media.tumblr.com/tumblr_j642n0xgph1r3nu00e1_500.png

bis zu zwei Hände und acht Finger



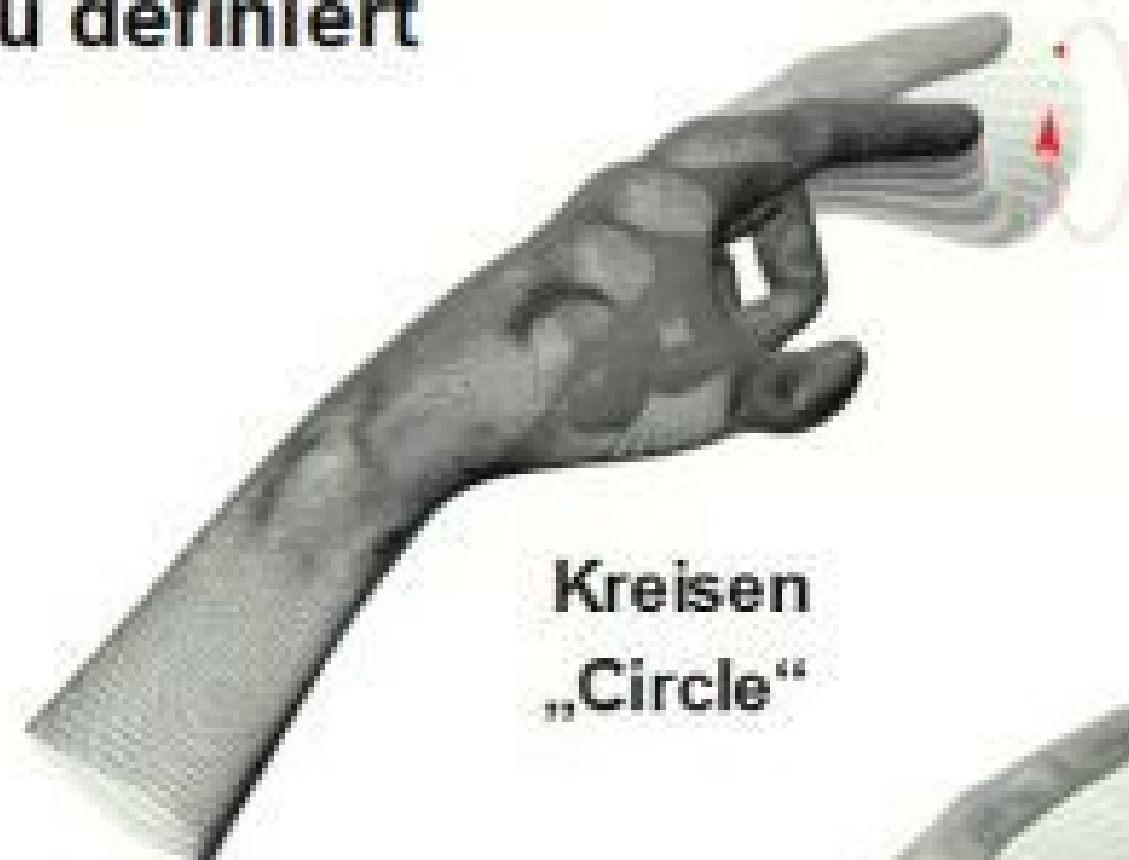
<http://quantum-box.com/box2/wp-content/uploads/2010/06/original.jpg>

Lage- und Positionsdaten

Mensch-Maschine Interaktion neu definiert



**Wischen
„Swipe“**



**Kreisen
„Circle“**



**Tastendruck
„Key Tap“**



**Bildschirmberührung
„Screen Tap“**



Stefan Wendler
Senior Software Consultant

stefan.wendler@tngtech.com



Dipl. Inf. (FH) Martin Förtsch
Senior Software Consultant

martin.foertsch@tngtech.com

- Martin Förtsch aka **Schwarzes Fragment**
 - <https://www.facebook.com/schwarzesfragment>



- Stefan Wendler aka **Binary Sun**
 - <http://www.binarysun.de>

